Das Gpg4win-Kompendium

Eine Veröffentlichung des Gpg4win-Projekts

Basierend auf einem Original von

Manfred J. Heinze, Karl Bihlmeier, Isabel Kramer
Dr. Francis Wray und Ute Bahn.

Überarbeitet von

Werner Koch, Emanuel Schütze, Dr. Jan-Oliver Wagner und Florian v. Samson.

Version 3.0.0-beta4 vom 28. September 2009

Impressum

[1] Wenn dieses Dokument kopiert, verteilt und/oder verändert wird, soll außer dieser Coypright-Notiz in keiner Form der Eindruck eines Zusammenhanges mit dem Bundesministerium für Wirtschaft und Technologie erweckt werden.

Über dieses Kompendium

Das Gpg4win-Kompendium besteht aus drei Teilen:

- **Teil I „Für Einsteiger“**: Der Schnelleinstieg in Gpg4win.
- **Teil II „Für Fortgeschrittene“**: Das Hintergrundwissen zu Gpg4win.
- **Anhang**: Weiterführende technische Informationen zu Gpg4win.

Teil I „Für Einsteiger“ führt Sie kurz und knapp durch die Installation und die alltägliche Benutzung der Gpg4win-Programmkomponenten. Der Übungsroboter **Adele** wird Ihnen dabei behilflich sein und ermöglicht Ihnen, die E-Mail-Ver- und Entschlüsselung (mit OpenPGP) so lange zu üben, bis Sie sich vertraut im Umgang mit Gpg4win gemacht haben.

Der Zeitbedarf für das Durcharbeiten des Schnelleinstiegs hängt unter anderem davon ab, wie gut Sie sich mit Ihrem PC und Windows auskennen. Sie sollten sich in etwa eine Stunde Zeit nehmen.

Teil II „Für Fortgeschrittene“ liefert Hintergrundwissen, das Ihnen die grundlegenden Mechanismen von Gpg4win verdeutlicht und die etwas seltener benutzten Fähigkeiten erläutert.

Teil 1 und 2 können unabhängig voneinander benutzt werden. Zu Ihrem besseren Verständnis sollten Sie aber möglichst beide Teile in der angegebenen Reihenfolge lesen.

Im **Anhang** finden Sie Details zu spezifischen technischen Themen rund um Gpg4win, unter anderem zur Outlook-Programmerweiterung GpgOL.

Wie das Kryptographie-Programmpaket Gpg4win selbst, wurde diese Dokument nicht für Mathematiker, Geheimdienstler und Kryptographen geschrieben, sondern für jedermann.

Legende

In diesem Kompendium werden folgende Textauszeichnungen benutzen:

- *Kursiv* wird dann verwendet, wenn etwas auf dem Bildschirm erscheint (z.B. in Menüs oder Dialogen). Zum Kennzeichnen von [*Schaltflächen*] werden zusäztlich eckige Klammern benutzt.

  Kursiv werden vereinzelt auch einzelne Wörter im Text gesetzt, wenn deren Bedeutung in einem Satz betont werden soll, das Schriftbild aber nicht durch die Auszeichnung „fett“ gestört werden soll (z.B.: *nur* OpenPGP).

- **Fett** wird für einzelne Wörter oder Sätze verwendet, die besonders wichtig und damit hervorzuheben sind. Diese Auszeichnung unterstützt dem Leser bei der schnelleren Erfassung hervorgehobener Schlüsselbegriffe und wichtiger Passagen.

- `Feste Laufweite` sind für alle Dateinamen, Pfadangaben, URLs, Quellcode sowie Ein- und Ausgaben (z.B. von Kommandozeilen) reserviert.

Inhaltsverzeichnis

Teil I

Für Einsteiger

1 Was ist Gpg4win?

Die Initiative Gpg4win (GNU Privacy Guard for Windows) ist eine Verschlüsselungssoftware für E-Mails und Dateien. Gpg4win bezeichnet ein **Gesamtpaket**, welches in Version 2 die folgenden Programme umfasst:

- **GnuPG**
  GnuPG ist das Kernstück von Gpg4win: Die Verschlüsselungs-Software.
- **Kleopatra**
  Die zentrale Zertifikatsverwaltung von Gpg4win. Unterstützt OpenPGP und X.509 (S/MIME) und sorgt für eine einheitliche Benutzerführung für alle kryptographischen Operationen.
- **GpgOL**
  GnuPG für Outlook (GpgOL) ist eine Erweiterung für Microsoft Outlook 2003 und 2007, die verwendet wird, um Nachrichten mit OpenPGP oder S/MIME zu signieren / verschlüsseln.
- **GpgEX**
  GPG Explorer eXtension (GpgEX) ist eine Erweiterung für den Windows Explorer, die es ermöglicht, Dateien über das Kontextmenü zu signieren / verschlüsseln.
- **GPA**
  Der GNU Privacy Assistent (GPA) ist neben Kleopatra ein alternatives Programm zum Verwalten von OpenPGP- und X.509-Zertifikaten.
- **Claws Mail**
  Claws Mail ist ein vollständiges E-Mail-Programm mit sehr guter Unterstützung für GnuPG.

Mit dem Verschlüsselungsprogramm GnuPG (GNU Privacy Guard) kann jedermann E-Mails sicher, einfach und kostenlos verschlüsseln. GnuPG kann ohne jede Restriktion privat oder kommerziell benutzt werden. Die von GnuPG eingesetzte Verschlüsselungstechnologie ist sicher und kann nach dem heutigen Stand von Forschung und Technik nicht gebrochen werden.

GnuPG ist **Freie Software**[1]. Das bedeutet, dass jedermann das Recht hat, sie nach Belieben kommerziell oder privat zu nutzen. Jedermann darf den Quellcode, also die eigentliche Programmierung des Programms, genau untersuchen und auch selbst Änderungen durchführen und diese weitergeben.[2]

Für eine Sicherheits-Software ist diese garantierte Transparenz des Quellcodes eine unverzichtbare Grundlage. Nur so lässt sich die Vertrauenswürdigkeit eines Programmes wirklich prüfen.

GnuPG basiert auf dem internationalen Standard **OpenPGP** (RFC 2440), ist vollständig kompatibel zu PGP und benutzt auch die gleiche Infrastruktur (Zertifikatsserver etc.) wie dieser. Seit Version 2

[1] oft auch als Open Source Software (OSS) bezeichnet

[2] Obwohl dies ausdrücklich erlaubt ist, sollte man ohne ausreichendes Fachwissen nicht leichtfertig Änderungen durchführen, da hierdurch die Sicherheit der Software beeinträchtigt werden kann.

von GnuPG wird auch der kryptographische Standard **S/MIME** (IETF RFC 3851, ITU-T X.509 und ISIS-MTT/Common PKI) unterstützt.

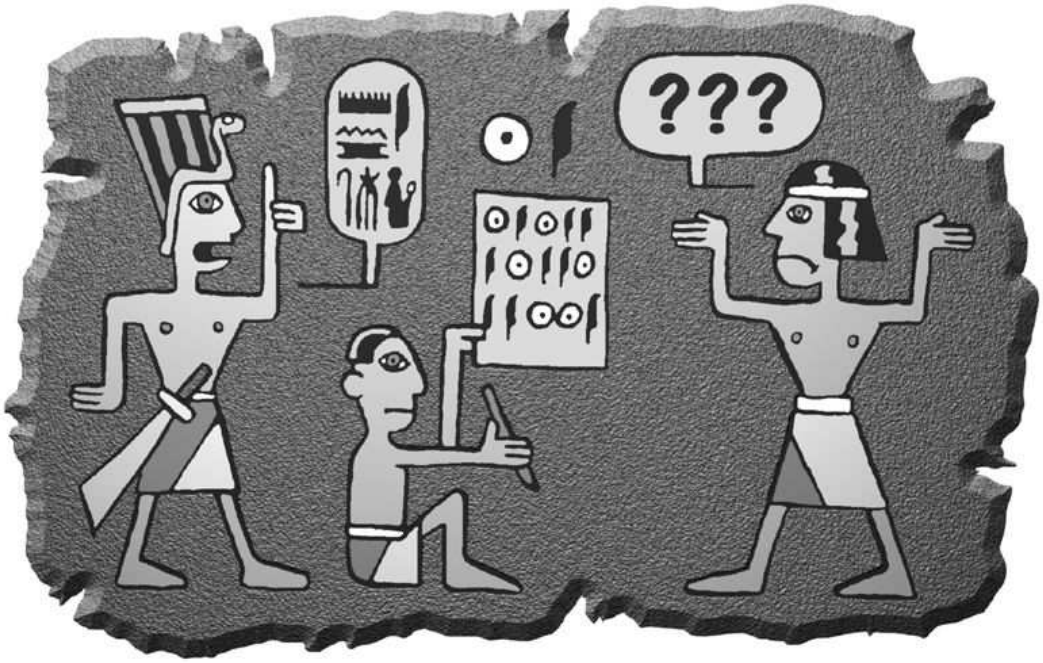
PGP („Pretty Good Privacy“) ist keine Freie Software, sie war lediglich vor vielen Jahren kurzzeitig zu ähnlichen Bedingungen wie GnuPG erhältlich. Diese Version entspricht aber schon lange nicht mehr dem Stand der Technik.

Die Vorläufer von Gpg4win wurden durch das Bundesministerium für Wirtschaft und Technologie, Gpg4win und Gpg4win2 durch das Bundesamt für Sicherheit in der Informationstechnik unterstützt.

Weitere Informationen zu GnuPG und weiteren Projekten der Bundesregierung zum Schutz des Internets finden Sie auf den Webseiten `www.bsi.de` und `www.bsi-fuer-buerger.de` des Bundesamtes für Sicherheit in der Informationstechnik.

2 Warum überhaupt verschlüsseln?

Die Verschlüsselung von Nachrichten wird manchmal als das zweitälteste Gewerbe der Welt bezeichnet. Verschlüsselungstechniken benutzten schon der Pharao Khnumhotep II, Herodot und Cäsar. Dank Gpg4win ist Verschlüsselung nunmehr für jedermann frei und kostenlos zugänglich.



Die Computertechnik hat uns phantastische Mittel in die Hand gegeben, um rund um den Globus miteinander zu kommunizieren und uns zu informieren. Aber Rechte und Freiheiten, die in anderen Kommunikationsformen längst selbstverständlich sind, muss man sich in den neuen Technologien erst sichern. Das Internet ist so schnell und massiv über uns hereingebrochen, dass man mit der Wahrung unserer Rechte noch nicht so recht nachgekommen sind.

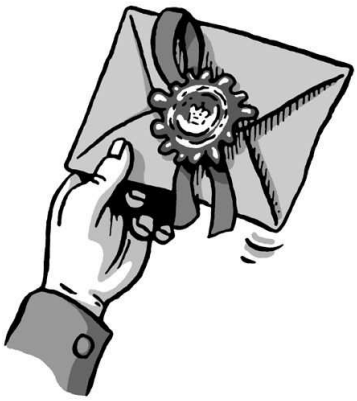
Beim altmodischen Briefschreiben schützen Sie die Inhalte von Mitteilungen ganz selbstverständlich mit einem Briefumschlag. Der Umschlag schützt die Nachrichten vor fremden Blicken, eine Manipulation am Umschlag kann man leicht bemerken. Nur wenn etwas nicht ganz so wichtig ist, schreibt man es auf eine ungeschützte Postkarte, die auch der Briefträger oder andere lesen können.

Ob die Nachricht wichtig, vertraulich oder geheim ist, das bestimmen Sie selbst und niemand sonst.

Diese Entscheidungsfreiheit haben Sie bei E-Mail nicht. Eine normale E-Mail ist immer offen wie eine Postkarte, und der elektronische „Briefträger“ – und andere – können sie immer lesen. Die Sache ist sogar noch schlimmer: Die Computertechnik bietet nicht nur die Möglichkeiten, die vielen Millionen E-Mails täglich zu befördern und zu verteilen, sondern auch, sie zu kontrollieren.

Niemand hätte je ernsthaft daran gedacht, alle Briefe und Postkarten zu sammeln, ihren Inhalt auszuwerten oder Absender und Empfänger zu protokollieren. Das wäre einfach nicht machbar gewesen, oder es hätte zu lange gedauert. Mit der modernen Computertechnik ist das technisch möglich. Es gibt mehr als einen Hinweis darauf, dass dies genau heute schon im großen Stil mit Ihrer E-Mail geschieht[1].

Denn: Der Umschlag fehlt.



[1] Hier sei nur an das `Echelon-System` erinnert (siehe `http://www.heise.de/tp/r4/artikel/6/6928/1.html`).

Vorschlag: Verwenden Sie einen „Umschlag“ für Ihre elektronischen Briefe. Ob Sie ihn benutzen, wann, für wen und wie oft, ist ganz allein Ihre Sache. Software wie Gpg4win gibt Ihnen lediglich die Wahlfreiheit zurück. Die Wahl, ob Sie persönlich eine Nachricht für wichtig und schützenswert halten oder nicht.

Das ist der Kern des Rechts auf Brief-, Post- und Fernmeldegeheimnis im Grundgesetz, und dieses Recht können Sie mit Hilfe des Softwarepakets Gpg4win wahrnehmen. Sie müssen diese Software nicht benutzen – Sie müssen ja auch keinen Briefumschlag benutzen. Aber es ist Ihr gutes Recht.

Um dieses Recht zu sichern, bietet Gpg4win Ihnen sogenannte „starke Verschlüsselungstechnik“. „Stark“ bedeutet hier: mit keinem gegenwärtigen Mittel zu knacken. In vielen Ländern waren starke Verschlüsselungsmethoden bis vor ein paar Jahren den Militärs und Regierungsbehörden vorbehalten. Das Recht, sie für jeden Bürger nutzbar zu machen, haben sich die Internetnutzer mühsam erobert; manchmal auch mit der Hilfe von klugen und weitsichtigen Menschen in Regierungsinstitutionen, wie im Falle der Portierung von GnuPG auf Windows. GnuPG wird von Sicherheitsexperten in aller Welt als eine praktikable und sichere Software angesehen.

Wie wertvoll diese Sicherheit für Sie ist, liegt ganz in Ihrer Hand.

Sie allein bestimmen das Verhältnis zwischen Bequemlichkeit bei der Verschlüsselung und größtmöglicher Sicherheit. Dazu gehören die wenigen, aber umso wichtigeren Vorkehrungen, die Sie treffen müssen, um Gpg4win richtig zu nutzen. In diesem Kompendium werden wir Ihnen dieses Vorgehen Schritt für Schritt erläutern.

3 Zwei Wege, ein Ziel: OpenPGP & S/MIME

Wie so oft gibt es für das gleiche Ziel verschiedene Wege, ähnlich ist es auch in der Verschlüsselung Ihrer E-Mails mit den Standards OpenPGP und S/MIME. Beide Standards und ihre Umsetzungen in Software ermöglichen die E-Mail-Verschlüsselung mit Freier Software, wie z.B. Gpg4win.

Beim Verschlüsseln bzw. bei der Sicherheit der geheimen Datenübertragung sind zwei Perspektiven wichtig, einmal die Gewährleistung der **Geheimhaltung** und zum anderen die **Authentizität** des Absenders. Authentizität bedeutet hier, dass der Inhalt auch tatsächlich vom besagten Absender versandt wurde.

Die Gemeinsamkeit: Das „Public-Key“ Verfahren

Konzeptionell steckt hinter OpenPGP und S/MIME die gleiche Methode zur Geheimhaltung, und zwar das „Public-Key“ Verfahren. Was heißt das?

Stellen Sie sich vor, die E-Mail oder die Datei sei in einer Truhe verschlossen. Im Gegensatz zu einem „normalen“ Schloss mit einem Schlüssel gibt es beim „Public-Key“ Verfahren zum Verschlüsseln / Entschlüsseln ein **Schlüsselpaar**. So gibt es einen beglaubigten Schlüssel zum Verschlüsseln (der „**öffentliche Schlüssel**“) und einen Schlüssel zum Entschlüsseln (der „**geheime Schlüssel**“).

Betrachtet man den öffentlichen Schlüssel zusammen mit Angaben über den Schlüssel (sogenannten Metadaten), so spricht man von einem öffentlichen **Zertifikat**. Metadaten können z.B. die zum Schlüssel zugehörige E-Mail-Adresse, die Benutzer-Kennung oder der Gültigkeitszeitraum sein (vgl. Kapitel 14).

Die Eigenschaften geheim / privat bzw. öffentlich sind völlig unabhängig (orthogonal) davon, ob sie sich auf einen reinen („nackten“) Schlüssel oder ein Zertifikat (Schlüssel mit Metadaten) beziehen.

Anmerkung: Technisch werden für die eigentliche Kryptographie (Verschlüsseln und Signieren) nur Schlüssel verwendet; praktisch handhabt man ausschließlich Zertifikate (z.B. in den Gpg4win-Programmkomponenten).

Bezogen auf das obige Beispiel mit der Truhe klingt es komisch, ein öffentlichen und geheimen Schlüssel zu haben. Aber bei Software löst diese Idee das Problem, dass man seinen Schlüssel für jeden Empfänger aus der Hand geben müsste. Denn normalerweise muss ein Schlüssel zum Verschlüsseln / Abschließen auch zum Entschlüsseln bzw. Aufschließen benutzt werden. Also müsste man Ihnen, wenn man etwas für Sie in der Truhe verschließt, die Truhe *und* den Schlüssel geben. Wenn der Schlüssel bei der Übertragung abhanden kommt oder jemand davon eine Kopie erstellt, ist das ein großes Problem.

Beim „Public-Key“ Verfahren verschließt man mit Ihrem **öffentlichen Schlüssel** (aus dem Zertifikat) die Truhe und Sie schließen die Truhe mit Ihrem **geheimen Schlüssel** auf. Man muss also nur die Truhe zu Ihnen transportieren lassen. Das ist auf jeden Fall sicherer als den geheimen Schlüssel mit zu transportieren, selbst wenn er einen anderen Weg als die Truhe zu Ihnen nehmen würde.

Trotz dieses gleichen Ansatzes zur Geheimhaltung unterscheiden sich OpenPGP und S/MIME aber z.B. bei der Schlüsselerzeugung (Näheres erfahren Sie später im Kapitel 5).

Falls Sie sich jetzt fragen, wie das „Public-Key“ Verfahren so funktionieren kann, lesen Sie einmal Kapitel 12. Wenn Sie sich dann noch fragen, warum Gpg4win so sicher ist, sind vermutlich die Kapitel 24 und 25 genau das richtige für Sie! Mit ein wenig Interesse und Zeit kann man dort auch die kleinen mathematischen Geheimnisse verstehen. Viel Spaß beim Entdecken.

Der Unterschied: Die Authentisierung / Beglaubigungen

Der wesentliche Unterschied zwischen OpenPGP und S/MIME liegt im Bereich der Authentisierung / Beglaubigungen.

Um die Authentizität des Absenders festzustellen, ist bei **S/MIME** ein Zertifikat notwendig, welches die Authentizität des Schlüsselpaar-Besitzers unzweifelhaft beglaubigt. Das heißt, dass Sie Ihren öffentlichen Schlüssel von einer dazu berechtigten Organisation beglaubigen lassen müssen, bevor er wirklich nutzbar wird. Das Zertifikat dieser Organisation wurde wiederum mit dem Zertifikat einer höher stehenden Organisation beglaubigt usw. bis man zu einem Wurzelzertifikat kommt. Vertraut man nun diesem Wurzelzertifikat, so vertraut man automatisch allen darunter liegenden Zertifikaten. Das nennt man **hierarchisches Vertrauenskonzept**. Zumeist ist die Kette nur 3 Komponenten lang: Wurzelzertifikat, Zertifikat des Zertifikatsausstellers (auch CA für Certificate Authority genannt), Anwenderzertifikat. Technisch ist eine Beglaubigung nichts anderes als eine Signatur des Beglaubigenden: hier wird also das aus der Zertifikatsanfrage erstellte x.509-Zertifikat von dem Zertifikatsaussteller signiert und diese Signatur an das Zertifikat angefügt.

Im Gegensatz dazu verwendet **OpenPGP** in der Regel eine direkte („peer-to-peer“) Beglaubigung (Anwender A beglaubigt Anwender B, B beglaubigt A und C usw.) und macht damit aus einem Beglaubigungs-Baum ein Beglaubigungs-Netz, das sogenannte **Web-of-Trust**. Im Fall der direkten Authentisierung bei OpenPGP haben Sie also die Möglichkeit, *ohne* die Beglaubigung von einer höheren Stelle, verschlüsselte Daten und E-Mails auszutauschen. Dafür reicht es aus, wenn Sie der E-Mail-Adresse und dem dazugehörigen Zertifikat ihres Kommunikationspartners vertrauen.

Nähere Informationen zu Authentisierungswegen, wie z.B. dem Web-of-Trust oder den Beglaubigungsinstanzen (CAs), erhalten Sie in Kapitel 16.

Kurz zusammengefasst

Was bedeutet das für Sie?

- Sowohl **OpenPGP** als auch **S/MIME** kann Ihnen die notwendige Sicherheit bieten.
- Beide Verfahren sind **nicht kompatibel** miteinander. Sie bieten zwei separate Wege bei der Authentisierung Ihrer geheimen Kommunikation. Man sagt, sie sind nicht interoperabel.
- **Gpg4win** als Freie Software ermöglicht Ihnen die bequeme **parallele** Nutzung beider Verfahren.

Falls Ihnen das etwas zuviel Informationen waren, machen Sie sich keine Sorgen: In den folgenden Kapiteln wird jeder Schritt von der Installation bis hin zur Verschlüsselung sowohl mit OpenPGP als auch mit S/MIME detailliert erklärt.

Die beiden nachfolgenden Symbole weisen Sie in diesem Kompendium auf spezifische Erklärungen zu OpenPGP bzw. S/MIME hin, so dass Sie immer schnell überblicken, welche Besonderheiten bei welchem Konzept zu beachten sind.

4 Sie installieren Gpg4win

Beginnen Sie nun mit der Installation von Gpg4win. Beachten Sie, dass Sie dafür Administratorrechte auf Ihrem Windows-Betriebssystem benötigen.

Sollte bereits eine GnuPG basierte Anwendung, wie z.B. GnuPP, GnuPT, WinPT oder GnuPG Basics, auf Ihrem Rechner installiert sein, so lesen sie jetzt bitte zuerst den Anhang E, um zu erfahren wie Sie Ihre vorhandenen Zertifikate übernehmen können.

Falls Sie Gpg4win aus dem Internet heruntergeladen haben:

Klicken Sie bitte auf diese neu abgespeicherte Datei, die folgenden Namen haben sollte: `gpg4win-2.0.0.exe` (oder mit einer höheren Versionsnummer).

Achten Sie unbedingt darauf, dass Sie die Datei von einer vertrauenswürdigen Seite erhalten haben, z.B.: `www.gpg4win.de`

Falls Sie Gpg4win auf einer CD-ROM erhalten haben:

Legen Sie diese CD-ROM in das CD-ROM-Laufwerk Ihres PCs. Öffnen Sie Ihren „Arbeitsplatz“ und klicken Sie dort auf das CD-ROM-Icon mit dem Titel „Gpg4win“. Anschließen klicken Sie auf das Installations-Icon mit dem Titel „Gpg4win“.

Die weitere Installation ist dann identisch:

Die Frage, ob Sie das Programm installieren wollen, beantworten Sie mit [*Ja*].

Der Installationsassistent startet und befragt Sie zuerst nach der Sprache für den Installationsvorgang:

Bestätigen Sie Ihre Sprachauswahl mit [*OK*].

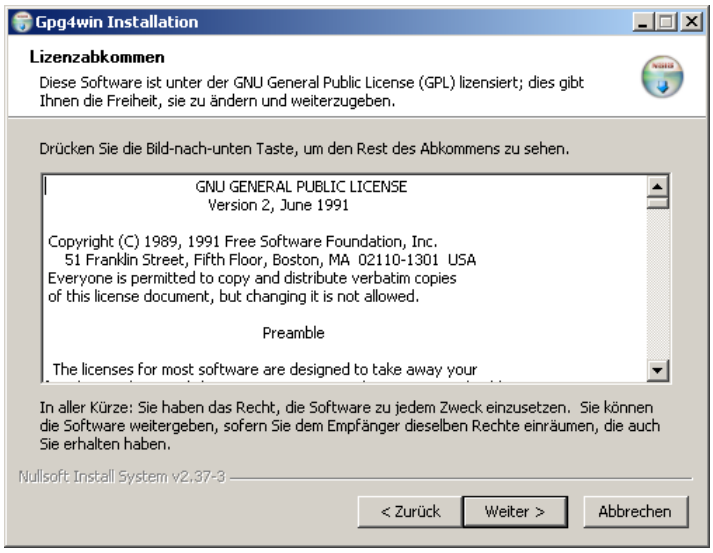
Anschließend begrüßt Sie dieser Willkommensdialog:

Beenden Sie alle auf Ihrem Rechner laufenden Programme, und klicken Sie dann auf [*Weiter*].

Auf der Seite mit dem **Lizenzabkommen**, können Sie Informationen zu den Lizenzen dieser Software lesen.

Wenn Sie die Software lediglich installieren und einsetzen wollen, so haben Sie immer das Recht dazu und sind nicht angehalten diese Texte zu lesen.

Geben Sie allerdings diese Software weiter oder wollen Sie sie verändern, so müssen Sie sich mit den Bedingungen der Lizenzen vertraut machen.



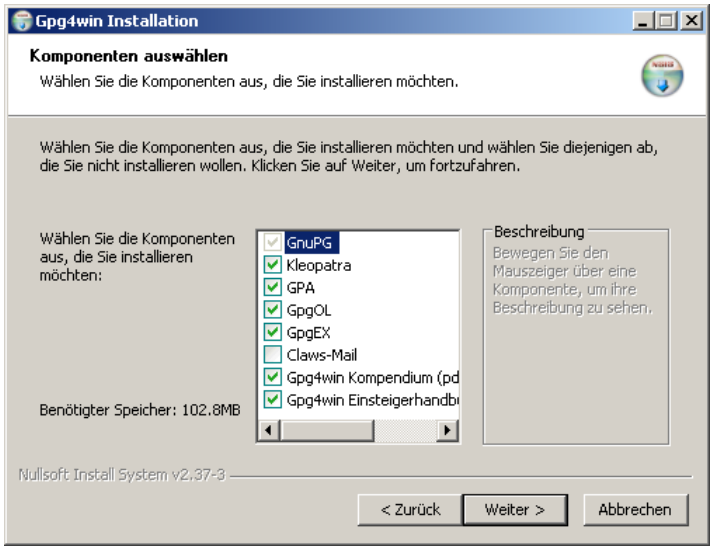
Klicken Sie auf [*Weiter*].

Auf der Seite mit der **Komponentenauswahl** können Sie entscheiden, welche Programme Sie installieren möchten.

Eine Vorauswahl der normalerweise installierende Komponenten ist bereits getroffen. Den Rest können Sie bei Bedarf auch später installieren.

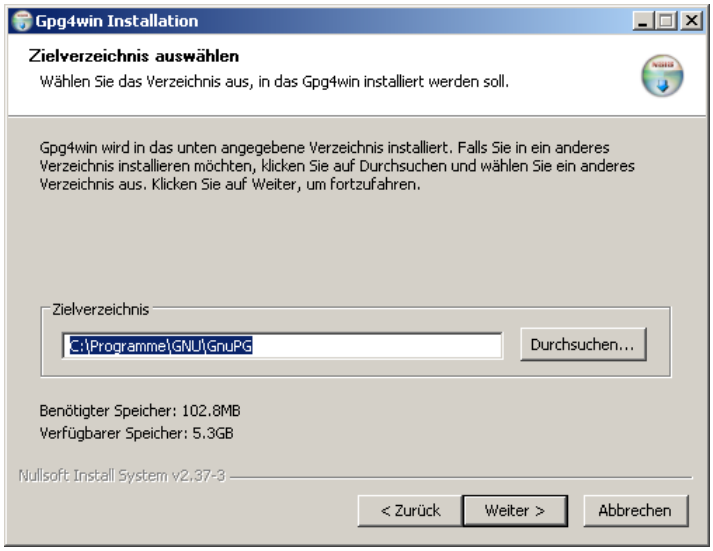
Wenn Sie die Maus über eine Gpg4win-Komponente ziehen, dann erscheint jeweils rechts eine Kurzbeschreibung die Ihnen bei der Entscheidung hilft.

Die Anzeige des benötigen Speichers auf der Festplatte von allen zur Installation ausgewählten Gpg4win-Komponenten kann auch sehr hilfreich sein.



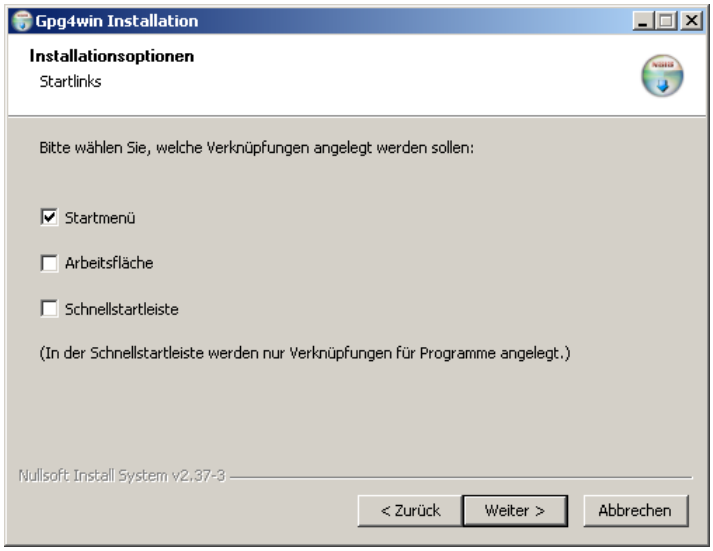
Klicken Sie auf [*Weiter*].

In der nun folgenden **Installationsverzeichnis-Auswahl** können Sie einen Dateiordner auf Ihrem PC aussuchen, in dem Gpg4win installiert wird. Sie können hier in der Regel den vorgeschlagenen Programm-Ordner übernehmen, z.B.: `C:\Programme\GNU\GnuPG`



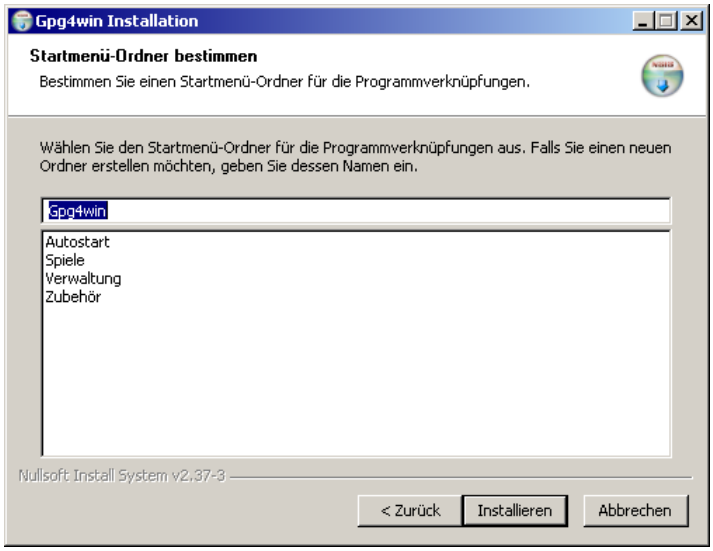
Klicken Sie anschließend auf [*Weiter*].

Auf der folgenden Seite können Sie festlegen, welche **Verknüpfungen** installiert werden. Voreingestellt ist lediglich eine Verknüpfung mit dem Startmenü. Diese Verknüpfungen können auch jederzeit später mit den Bordmitteln von Windows verändert werden.



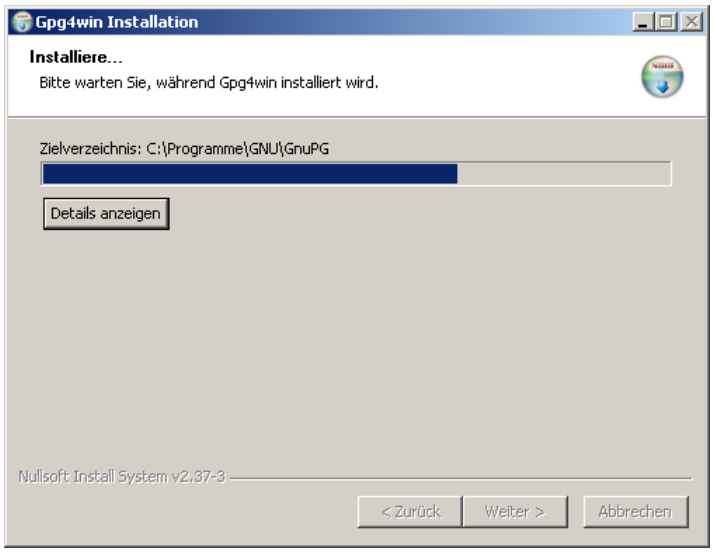
Klicken Sie anschließend auf [*Weiter*].

Falls Sie auf der vorhergehenden Seite eine **Verknüpfung mit dem Startmenü** ausgewählt haben (dies ist die Voreinstellung), so wird Ihnen nun eine Seite angezeigt, mit der Sie den Namen dieses Startmenüs festlegen können.



Am einfachsten übernehmen Sie den vorgeschlagenen Namen und klicken dann auf [*Installieren*].

Während der nun folgenden **Installation** sehen Sie einen Fortschrittsbalken und Informationen, welche Datei momentan installiert wird. Sie können jederzeit auf [*Details anzeigen*] drücken, um ein Protokoll der Installation sichtbar zu machen.



Nachdem die Installation abgeschlossen ist, drücken Sie bitte auf [*Weiter*].

Nach erfolgreicher Installation wird Ihnen diese letzte Seite des Installationsvorgangs angezeigt:

Sofern Sie die README-Datei nicht ansehen wollen, deaktivieren Sie die Option auf dieser Seite. Klicken Sie schließlich auf [*Fertig stellen*].

In einigen Fällen kann es vorkommen, dass Windows neu gestartet werden muss. In diesem Fall sehen Sie statt der vorherigen die folgende Seite:

Sie können hier auswählen, ob Windows sofort oder später manuell neu gestartet werden soll.

Klicken Sie auf [*Fertig stellen*].

Das war's schon!

Sie haben Gpg4win erfolgreich installiert und können es gleich zum ersten Mal starten.

Vorher sollten Sie aber Kapitel 12 lesen. Dort erfahren Sie den genialen Trick, mit dem Gpg4win Ihre E-Mails sicher und bequem verschlüsselt. Gpg4win funktioniert zwar auch, ohne dass Sie verstehen warum, aber im Gegensatz zu anderen Programmen wollen Sie Gpg4win schließlich Ihre geheime Korrespondenz anvertrauen. Da sollten Sie schon wissen, was vor sich geht.

Außerdem ist die ganze Angelegenheit ziemlich spannend. . .

In Kapitel 13 bekommen Sie einige Tipps, mit denen Sie sich einen sicheren und trotzdem leicht zu merkenden Passphrase ausdenken können.

Für Informationen zur **automatischen Installation** von Gpg4win (wie sie z.B. für Softwareverteilungs-Systeme interessant ist), lesen Sie bitte im Anhang D „Automatische Installation von Gpg4win“ weiter.

5 Sie erzeugen Ihr Schlüsselpaar

Spätestens nachdem Sie gelesen haben, warum GnuPG eigentlich so sicher ist (Kapitel 24) und wie eine gute Passphrase als Schutz für Ihren geheimen Schlüssel entsteht (Kapitel 13), möchten Sie nun Ihr persönliches Schlüsselpaar erzeugen.

Ein Schlüsselpaar besteht, wie Sie im Kapitel 3 gelernt haben, aus einem **öffentlichen** und einem **geheimen Schlüssel**. Ergänzt mit den Metadaten (E-Mail-Adresse, Benutzer-Kennung etc.), die Sie bei der Erstellung Ihres Schlüsselpaars angeben, erhalten Sie Ihr Zertifikat mit dem öffentlichen und geheimen Schlüssel.

Diese Definition gilt sowohl für OpenPGP wie auch für S/MIME (die Zertifikate entsprechen einem Standard mit der Bezeichnung „X.509“).

Eigentlich müsste man diesen wichtigen Schritt der Schlüsselpaar-Erzeugung ein paar Mal üben können. . .

Genau das können Sie tun – und zwar für OpenPGP:

Sie können den gesamten Ablauf der Schlüsselpaar-Erzeugung, Verschlüsselung und Entschlüsselung durchspielen, so oft Sie wollen, bis Sie ganz sicher sind.

Ihr Vertrauen in Gpg4win wird sich durch diese „Trockenübung“ festigen, und die „heisse Phase“ der OpenPGP-Schlüsselpaar-Erzeugung wird danach kein Problem mehr sein.

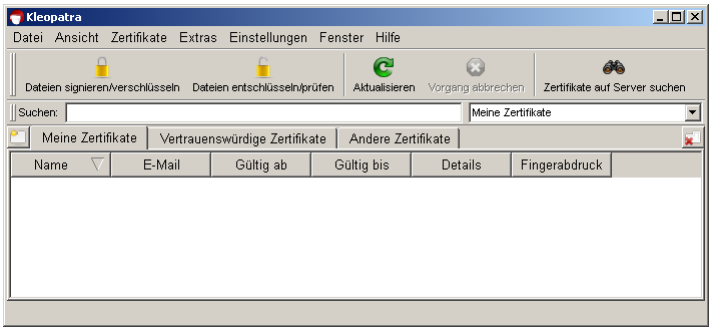
Ihr Partner bei diesen Übungen wird **Adele** sein.

Adele ist ein Testservice, der noch aus dem alten GnuPP-Projekt stammt und ist bis auf weiteres noch in Betrieb. „Das Gpg4win-Kompendium“ verwendet diesen zuverlässigen Übungsroboter und dankt den Inhabern von gnupp.de für den Betrieb von Adele.

Mit Hilfe von Adele können Sie Ihr OpenPGP-Schlüsselpaar, das Sie gleich erzeugen werden, ausprobieren und testen, bevor Sie damit Ernst machen. Doch dazu später mehr.

Los geht's! Rufen Sie das Programm Kleopatra über das Windows-Startmenü auf:

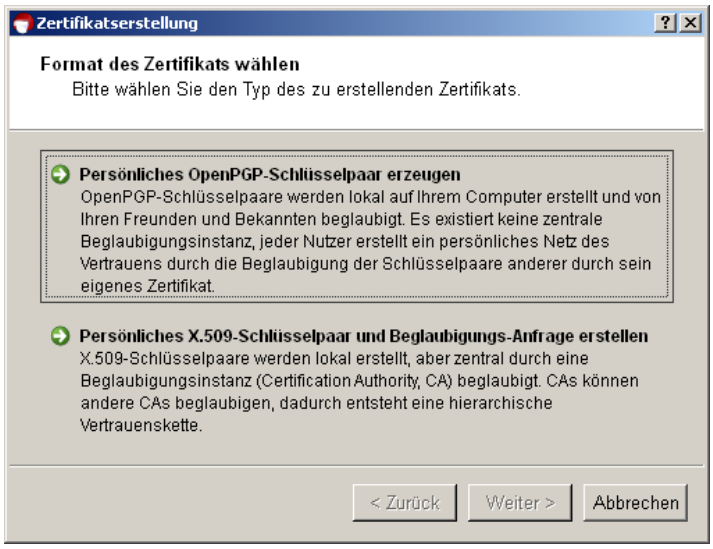
Daraufhin sehen Sie das Hauptfenster von Kleopatra – die Zertifikatsverwaltung:



Zu Beginn ist diese Übersicht leer, da Sie noch keine Zertifikate erstellt (oder importiert) haben. Dies können Sie jetzt nachholen...

Klicken Sie auf *Datei→Neues Zertifikat*.

Im folgenden Dialog entscheiden Sie sich für ein Format, in dem anschließend ein Zertifikat erstellt werden soll. Sie haben die Wahl zwischen **OpenPGP** (PGP/MIME) oder **X.509** (S/MIME). Zu deren Unterschieden lesen Sie bitte Kapitel 3.



Die weitere Vorgehensweise zum Erzeugen eines Schlüsselpaars gliedert sich an dieser Stelle in zwei Abschnitte:

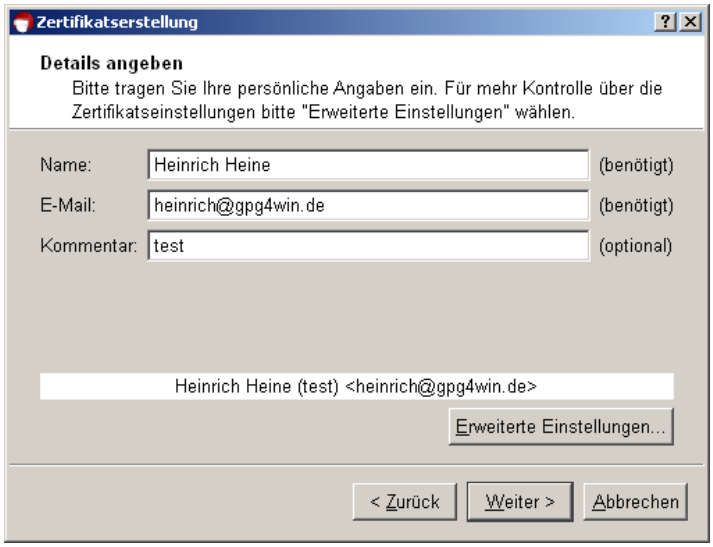
- Abschnitt 5.1: **OpenPGP-Schlüsselpaar erstellen** (siehe nächste Seite) und
- Abschnitt 5.2: **X.509-Schlüsselpaar erstellen** (siehe Seite 37).

Lesen Sie den entsprechenden Abschnitt weiter, je nachdem für welches Zertifikatsformat Sie sich entschieden haben.

5.1 OpenPGP-Schlüsselpaar erstellen

Klicken Sie im obigen Auswahldialog auf die Schaltfläche [*Persönliches OpenPGP-Schlüsselpaar erzeugen*].

Geben Sie im nun folgenden Fenster Ihren Namen und Ihre E-Mail-Adresse an.



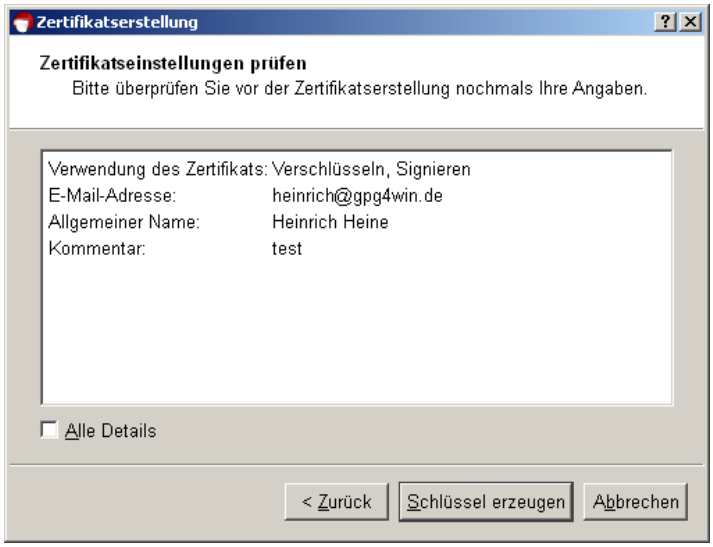
Wenn Sie die OpenPGP-Schlüsselpaar-Erzeugung zunächst einmal **testen** wollen, dann können Sie einfach einen beliebigen Namen und irgendeine ausgedachte E-Mail-Adresse eingeben, z.B.: `Heinrich Heine` und `heinrichh@gpg4win.de`.

Optional können Sie einen Kommentar zum Schlüsselpaar eingeben. Normalerweise bleibt dieses Feld leer; wenn sie aber einen Testschlüssel erzeugen, sollten Sie dort als Erinnerung „test“ eingeben. Dieser Kommentar ist Teil Ihrer User-ID und genau wie der Name und die E-Mail-Adresse später öffentlich sichtbar.

Die **erweiterten Einstellungen** benötigen Sie nur in Ausnahmefällen. Sie können sich im Kleopatra Handbuch (über *Hilfe→Kleopatra Handbuch*) über die Details informieren.

Klicken Sie auf [*Weiter*].

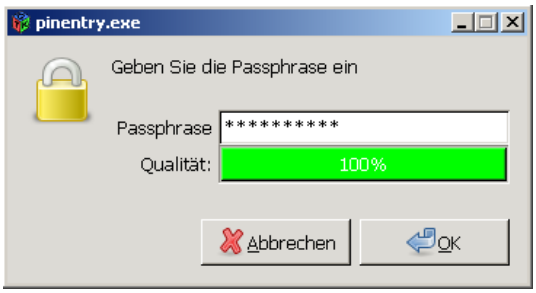
Es werden nun noch einmal alle wesentlichen Eingaben und Einstellungen zur **Kontrolle** aufgelistet. Falls Sie sich für die (voreingestellten) Experten-Einstellungen interessieren, können Sie diese über die Option *Alle Details* einsehen.



Sofern alles korrekt ist, klicken Sie anschließend auf [*Schlüssel erzeugen*].

Jetzt folgt der wichtigste Teil: Die Eingabe Ihrer **Passphrase**!

Während der Schlüsselgenerierung werden Sie aufgefordert Ihre persönliche Passphrase einzugeben:



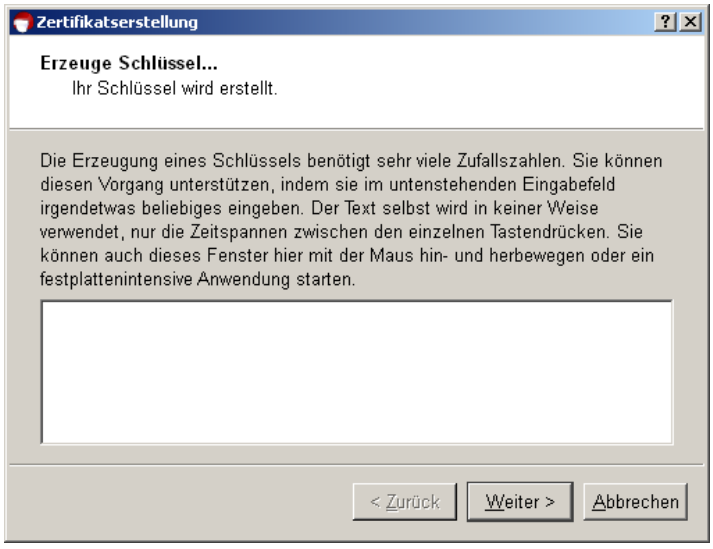
Im Kapitel 13 (Seite 89) erhalten Sie einige wertvolle Tipps, was Sie bei der Erzeugung einer **sicheren Passphrase** beachten sollten. Nehmen Sie die Sicherheit Ihrer Passphrase ernst!

Sie sollten nun eine geheime, einfach zu merkende und schwer zu knackende Passphrase parat haben und im obigen Dialog eintragen.

Auch an dieser Stelle können Sie – wenn Sie wollen – zunächst eine **Test-Passphrase** eingeben oder auch gleich „Ernst machen“.

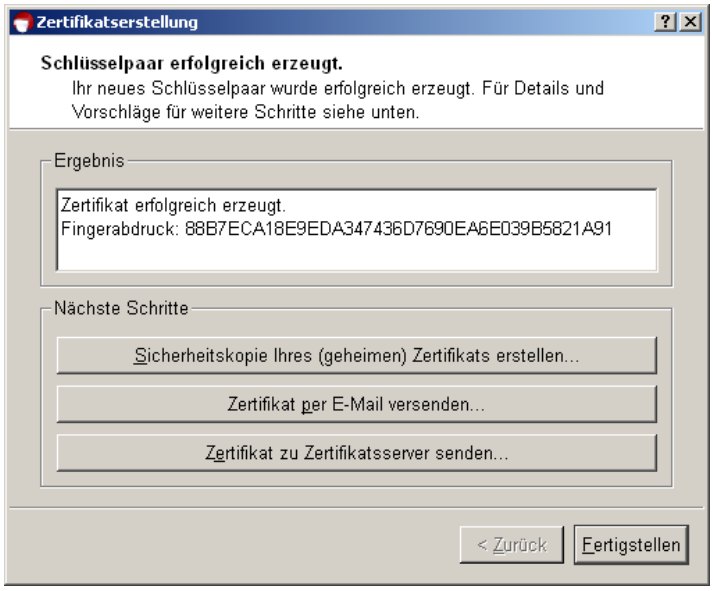
Um sicher zu gehen, dass Sie sich nicht vertippen, müssen Ihre geheime Passphrase zweimal eingeben. Bestätigen Sie Ihre Eingabe jeweils mit [*OK*].

Nun wird Ihr OpenPGP-Schlüsselpaar angelegt:



Dies kann u.U. einige Minuten dauern. Sie können in der Zwischenzeit mit einer anderen Anwendung Ihres Rechner weiterarbeiten und erhöhen hierdurch sogar leicht die Qualität des erzeugten Schlüsselpaars.

Sobald die **Schlüsselpaargenerierung erfolgreich** abgeschlossen ist, erhalten Sie folgenden Dialog:



Im Ergebnis-Textfeld wird der 40-stellige „Fingerabdruck“ Ihres neu generierten OpenPGP-Zertifikats angezeigt. Dieser Fingerabdruck (engl. „Fingerprint“) ist weltweit eindeutig, d.h. keine andere Person besitzt ein Zertifikat mit identischem Fingerabdruck. Es ist sogar vielmehr so, dass es schon mit 8 Zeichen ein außerordentlicher Zufall wäre, wenn diese weltweit ein zweites Mal vorkämen. Daher werden oft nur die letzten 8 Zeichen des Fingerabdrucks verwendet bzw. angezeigt. Dieser Fingerabdruck identifiziert die Identität des Zertifikats wie der Fingerabdruck einer Person.

Sie brauchen sich den Fingerabdruck nicht zu merken oder abzuschreiben. In den Zertifikatsdetails von Kleopatra können Sie sich diese jederzeit später anzeigen lassen.

Als nächstes können Sie eine oder mehrere der folgenden drei Schaltflächen betätigen:

Sicherheitskopie Ihres (geheimen) Zertifikats erstellen...

Geben Sie hier den Pfad an, wohin Ihr vollständiges Zertifikat (das Ihr neues Schlüsselpaar enthält) exportiert werden soll:

Kleopatra wählt automatisch den Dateityp und speichert Ihr Zertifikat als `.asc` bzw. `.gpg` Datei ab – abhängig davon, ob Sie die Option **ASCII-geschützt** (engl. „ASCII armor“) ein- bzw. ausschalten.

Klicken Sie anschließend zum Exportieren auf [*OK*].

Wichtig: Falls Sie die Datei auf der Festplatte abspeichern, so sollten Sie baldmöglichst diese Datei auf einen anderen Datenträger (USB Stick, Diskette oder CDROM) kopieren und diese Orginaldatei rückstandslos löschen (nicht im Papierkorb belassen). Bewahren Sie diesen Datenträger sicher auf.

Sie können eine Sicherheitskopie auch jederzeit später anlegen; wählen Sie hierzu aus dem Kleopatra-Hauptmenü: *Datei→Geheimes Zertifiakt exportieren...* (vgl. Kapitel 19).

Zertifikat per E-Mail versenden...

Nach Drücken dieser Schaltfläche sollte eine neue E-Mail erstellt werden – mit Ihrem neuen öffentlichen Zertifikat im Anhang. Ihr geheimer OpenPGP-Schlüssel wird selbstverständlich *nicht* versendet. Geben Sie eine Empfänger-E-Mail-Adresse an und ergänzen Sie ggf. den vorbereiteten Text dieser E-Mail.

Beachten Sie: Nicht alle E-Mail-Programme unterstützen diese Funktion. Sollte kein neues E-Mail-Fenster aufgehen, so beenden Sie den Zertifikatserstellungs-Assistenten, speichern Ihr öffentliches Zertifikat durch *Datei→Zertifikat exportieren* und versenden diese Datei per E-Mail an Ihre Korrespondenzpartner (Details im Abschnitt 6.1).

Zertifikate zu Zertifikatsserver senden...

Wie genau Sie einen weltweit verfügbaren OpenPGP-Zertifikatsserver in Kleopatra einrichten und wie Sie anschließend Ihr öffentliches Zertifikat auf diesen Server veröffenlichen, erfahren Sie in Kapitel 15.

Beenden Sie anschließend den Kleopatra-Assistenten mit [*Fertigstellen*], um die Erzeugung Ihres OpenPGP-Zertifikats abzuschließen.

Weiter geht’s mit dem Abschnitt „Schlüsselpaar-Erzeugung abgeschlossen“ auf Seite 43. Von da an sind die Erklärungen für OpenPGP und X.509 wieder identisch.

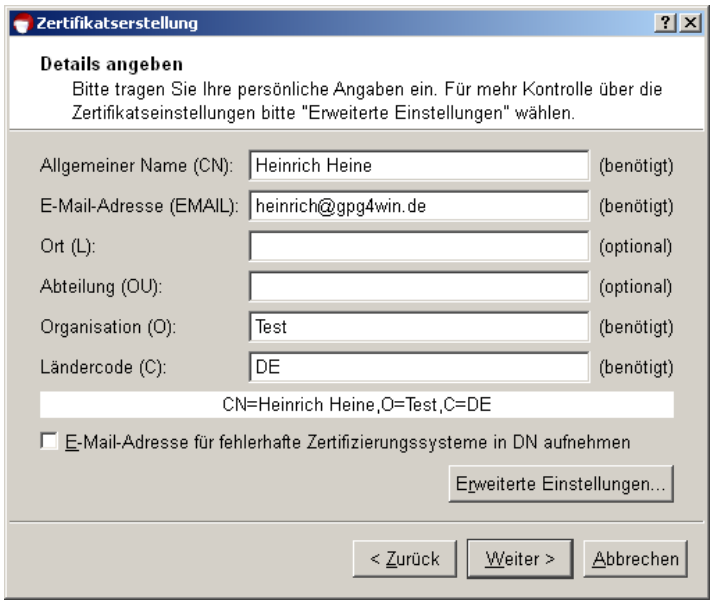
5.2 X.509-Schlüsselpaar erstellen

S/MIME

Klicken Sie im Zertifikatsformat-Auswahldialog von Seite 31 auf die Schaltfläche [*Persönliches X.509-Schlüsselpaar und Beglaubigungs-Anfrage erstellen*].

Geben Sie im nun folgenden Fenster Ihren Namen (CN), Ihre E-Mail-Adresse (EMAIL), Ihre Organisation (O) und Ihren Ländercode (C) an. Optional können Sie noch Ort (L) und Abteilung (OU) ergänzen.

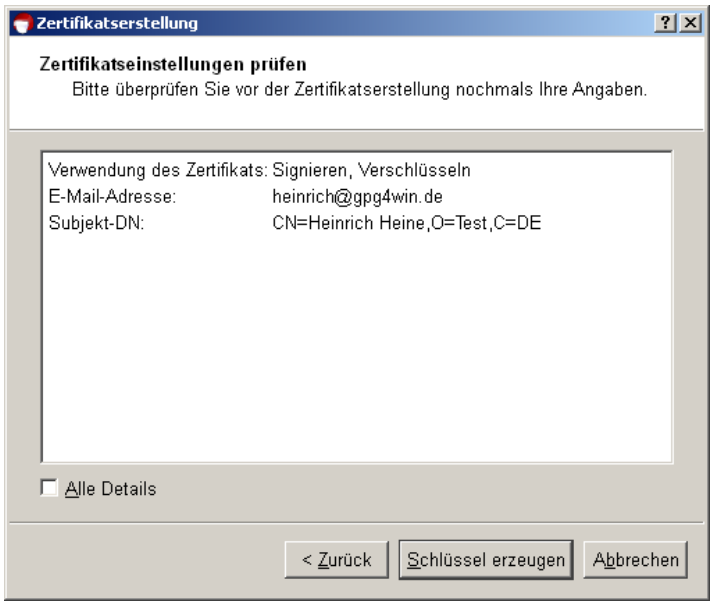
Wenn Sie die X.509-Schlüsselpaar-Erzeugung zunächst einmal **testen** wollen, dann machen Sie beliebige Angaben für Name, Organisation und Ländercode sowie geben irgendeine ausgedachte E-Mail-Adresse ein, z.B. `CN=Heinrich Heine,O=Test,C=DE,EMAIL=heinrichh@gpg4win.de`.



Die **erweiterten Einstellungen** benötigen Sie nur in Ausnahmefällen. Sie können sich im Kleopatra Handbuch (über *Hilfe→Kleopatra Handbuch*) über die Details informieren.

Klicken Sie auf [*Weiter*].

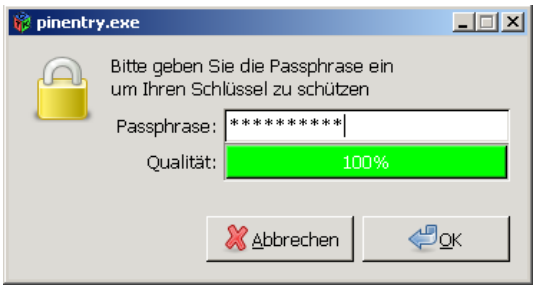
Es werden nun noch einmal alle Eingaben und Einstellungen zur **Kontrolle** aufgelistet. Falls Sie sich für die (voreingestellten) Experten-Einstellungen interessieren, können Sie diese über die Option *Alle Details* einsehen.



Sofern alles korrekt ist, klicken Sie anschließend auf [*Schlüssel erzeugen*].

Jetzt folgt der wichtigste Teil: Die Eingabe Ihrer **Passphrase**!

Während der Schlüsselgenerierung werden Sie aufgefordert Ihre Passphrase einzugeben:



Beachten Sie, dass dieses Pinentry-Fenster unter Umständen im Hintergrund geöffnet werden könnte (und damit auf dem ersten Blick nicht sichtbar ist).

Im Kapitel 13, Seite 89, erhalten Sie wertvolle Tipps, was Sie bei der Erzeugung einer **sicheren Passphrase** beachten sollten. Nehmen Sie die Sicherheit Ihrer Passphrase ernst!

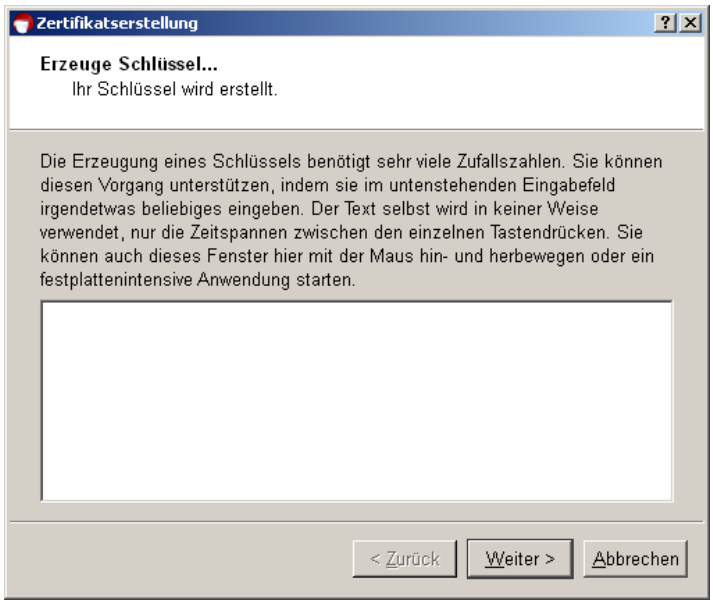
Sie sollten nun eine geheime, einfach zu merkende und schwer zu knackende Passphrase parat haben und im obigen Dialog eintragen.

Falls die Passphrase nicht sicher genug sein sollte (z.B. weil sie zu kurz ist oder keine Zahlen / Sonderzeichen enthält), werden Sie darauf hingewiesen.

Auch an dieser Stelle können Sie – wenn Sie wollen – zunächst eine **Test-Passphrase** eingeben oder auch gleich „Ernst machen“.

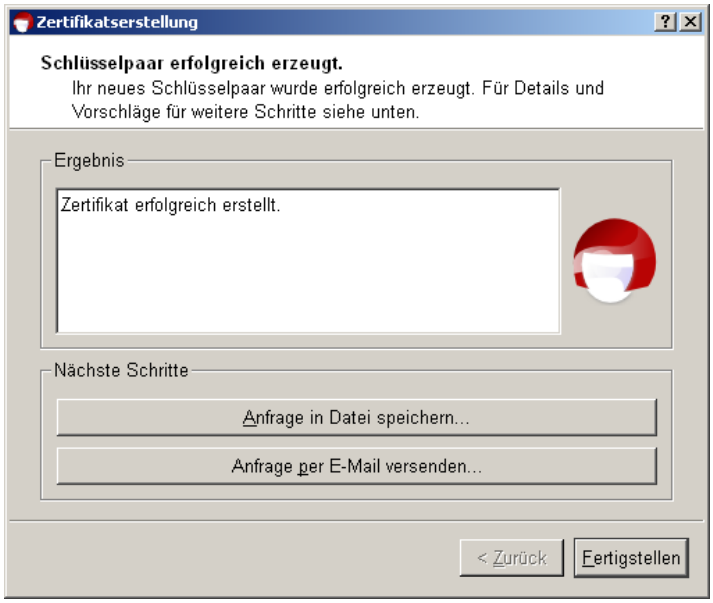
Um sicher zu gehen, dass Sie sich nicht vertippen, müssen Ihre geheime Passphrase zweimal eingeben. Abschließend werden Sie noch ein drittes Mal aufgefordert Ihre Passphrase einzugeben, um Ihre Zertifikatsanfrage mit Ihrem neuen geheimen Schlüssel zu signieren. Bestätigen Sie Ihre Eingaben jeweils mit [*OK*].

Nun wird Ihr X.509-Schlüsselpaar angelegt:



Dies kann u.U. einige Minuten dauern. Sie können in der Zwischenzeit mit einer anderen Anwendung Ihres Rechner weiterarbeiten und erhöhen hierdurch sogar leicht die Qualität des erzeugten Schlüsselpaars.

Sobald die **Schlüsselpaargenerierung erfolgreich** abgeschlossen ist, erhalten Sie folgenden Dialog:



Als nächstes können Sie eine oder mehrere der folgenden drei Schaltflächen betätigen:

Anfrage in Datei speichern...
: Geben Sie den genauen Pfad an, wohin Ihre X.509-Zertifikatsanfrage gespeichert werden soll und bestätigen Sie Ihre Eingabe. Kleopatra fügt beim Speichern automatisch die Dateiendung `.p10` hinzu. Sie können diese Datei dann später auf verschiedene Weise an eine Beglaubigungsinstanz geben.

Anfrage per E-Mail versenden...
: Es wird eine neue E-Mail erstellt – mit der soeben erstellen Zertifikatsanfrage im Anhang. Geben Sie eine Empfänger-E-Mail-Adresse an (in der Regel die Ihrer zuständigen Beglaubigungsinstanz (CA)) und ergänzen Sie ggf. den vorbereiteten Text dieser E-Mail.

  Beachten Sie: Nicht alle E-Mail-Programme unterstützen diese Funktion. Sollte kein neues E-Mail-Fenster aufgehen, so speichern Sie Ihre Anfrage zunächst in eine Datei (siehe oben) und versenden diese Datei per E-Mail an Ihre Beglaubigungsinstanz.

  Sobald die Anfrage von der CA bestätigt wurde, erhalten Sie von Ihrem zuständigen CA-Systemadministrator das fertige und unterzeichnete X.509-Zertifikat Ihres Schlüsselpaars. Dieses müssen Sie dann nur noch in Kleopatra importieren (vgl. Kapitel 19).

Beenden Sie anschließend den Kleopatra-Assistenten mit [*Fertigstellen*].

Erstellung eines X.509-Schlüsselpaars mit www.cacert.org

CAcert ist eine gemeinschaftsbetriebene, nicht-kommerzielle Beglaubigungsinstanz (CA), die kostenlos X.509-Zertifikate ausstellt.

Damit Sie sich ein (Client-)Zertifikat bei CAcert erstellen können, müssen Sie sich zunächst unter `www.cacert.org` registrieren.

Anschließend können Sie sich mit Ihrem CAcert-Account ein (oder mehrere) Client-Zertifikat(e) erstellen: Sie sollten dabei auf eine ausreichende Schlüssellänge (z.B. 2048 Bit) achten. In dem startenden Assistenten legen Sie Ihre sichere Passphrase für Ihr Zertifikat fest.

Ihre X.509-Zertifikatsanfrage wird nun erstellt.

Im Anschluss daran erhalten Sie eine E-Mail mit zwei Links zu Ihrem neu erstellten X.509-Zertifikat und dem dazugehörigen CAcert-Root-Zertifikat. Laden Sie sich beide Zertifikate herunter.

Folgen Sie den Anweisungen und installieren Ihr Zertifikat mit Ihrem Browser. Mit Firefox können Sie danach z.B. über *Bearbeiten→Einstellungen→Erweitert→Zertifikate* Ihr installiertes Zertifikat unter dem ersten Reiter „Ihre Zertifikate“ mit dem Namen (CN) **CAcert WoT User** finden.

Anmerkung: Sie können auch ein personalisiertes Zertifikat ausstellen, das z.B. Ihren Namen (im CN-Feld) trägt. Dazu müssen Sie sich mit Ihrem CAcert-Account von anderen Mitglieder bestätigen lassen. Das sogenannte „CACert-Web-of-Trust“ wächst dadurch. Was Sie für so eine Bestätigung tun müssen, erfahren Sie auf den Internetseiten von CAcert.

Speichern Sie abschließend eine Sicherungskopie Ihres X.509-Schlüsselpaars in einem X.509-Zertifikat (`.p12` Datei).
Achtung: Diese `.p12` Datei enthält Ihren öffentlichen und Ihren zugehörigen geheimen Schlüssel. Achten Sie darauf, dass diese Datei nicht in unbefugte Hände gelangt.

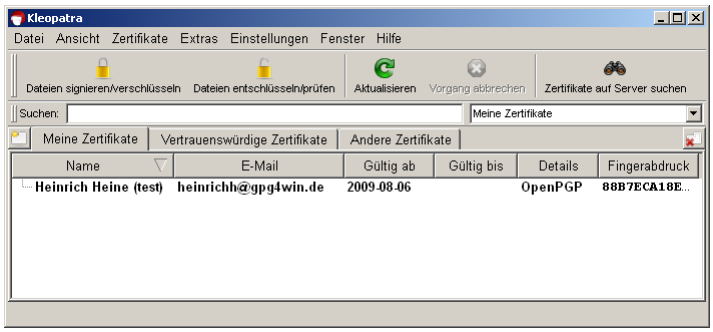
Wie Sie Ihr privates X.509-Zertifikat in Kleopatra importieren erfahren Sie in Kapitel 19.

Weiter geht’s mit Abschnitt 5.3 auf der nächsten Seite. Von nun an sind die Erklärungen für OpenPGP und X.509 wieder identisch.

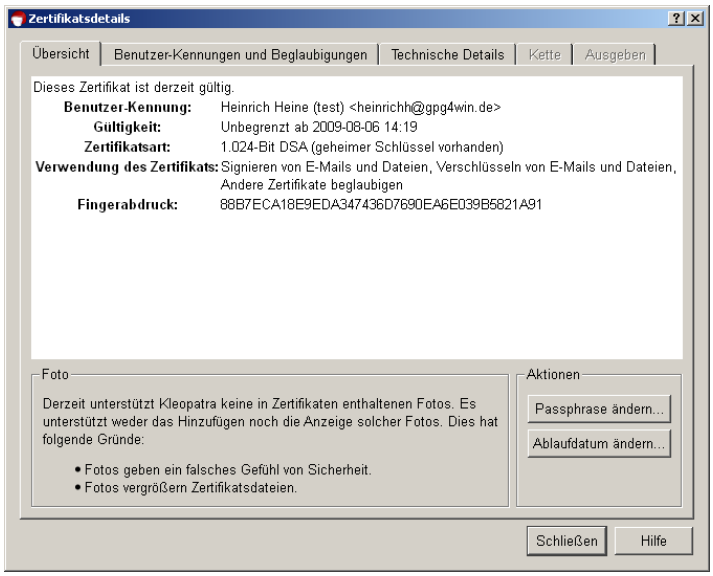
5.3 Schlüsselpaar-Erstellung abgeschlossen

Damit ist die Erzeugung Ihres OpenPGP- bzw. X.509-Schlüsselpaares abgeschlossen. Sie besitzen nun einen einzigartigen elektronischen Schlüssel.

Sie sehen jetzt wieder das Hauptfenster von Kleopatra. Das soeben erzeugte OpenPGP-Schlüsselpaar finden Sie in der Zertifikatsverwaltung unter dem Reiter *Meine Zertifikate* (hier und im weiteren wird exemplarisch ein OpenPGP-Zertifikat verwendet):



Doppelklicken Sie auf Ihr neues Zertifikat, um alle Zertifikatsdetails sehen zu können:



Was bedeuten die einzelnen Zertifikatsdetails?

Ihr Zertifikat ist unbegrenzt gültig, d.h. es hat kein „eingebautes Verfallsdatum“. Um die Gültigkeit nachträglich zu verändern, klicken Sie auf [*Ablaufdatum ändern*].

Weitere Informationen zu den Zertifikatsdetails finden Sie im Kapitel 14. Sie können dieses Kapitel jetzt lesen oder später, wenn Sie diese Informationen benötigen.

6 Sie veröffentlichen Ihr öffentliches Zertifikat

Beim täglichen Gebrauch von Gpg4win ist es sehr praktisch, dass Sie es beim Verschlüsseln und Signaturprüfen stets nur mit „ungeheimen“ (also öffentlichen) Zertifikaten zu tun haben, die nur öffentliche Schlüssel enthalten. Solange Ihr eigener geheimer Schlüssel und die ihn schützende Passphrase sicher sind, haben Sie das Wichtigste zur Geheimhaltung bereits erledigt.

Jedermann darf und soll Ihr öffentliches Zertifikat haben, und Sie können und sollen öffentliche Zertifikate von Ihren Korrespondenzpartnern haben – je mehr, desto besser.

Denn:

Um sichere E-Mails austauschen zu können, müssen beide Partner jeweils das öffentliche Zertifikat des anderen besitzen und benutzen. Natürlich braucht der Empfänger auch ein Programm, das mit Zertifikaten umgehen kann, wie z.B. das Softwarepaket Gpg4win mit der Zertifikatsverwaltung Kleopatra.

Wenn Sie also an jemanden verschlüsselte E-Mails schicken wollen, müssen Sie dessen öffentliches Zertifikat haben und zum Verschlüsseln benutzen.

Wenn – andersherum – jemand Ihnen verschlüsselte E-Mails schicken will, muss er Ihr öffentliches Zertifikat haben und zum Verschlüsseln benutzen.

Deshalb sollten Sie nun Ihr öffentliches Zertifikat zugänglich machen. Je nachdem, wie groß der Kreis Ihrer Korrespondenzpartner ist und welches Zertifikatsformat Sie einsetzen, gibt es verschiedene Möglichkeiten. Verbreiten Sie Ihr öffentliches Zertifikat beispielsweise...

- ... direkt per **E-Mail** an bestimmte Korrespondenzpartner (vgl. Abschnitt 6.1).
- ... auf einem **OpenPGP-Zertifikatsserver**; gilt *nur* für OpenPGP (vgl. Abschnitt 6.2).
- ... über die eigene Homepage.
- ... persönlich, z.B. per USB-Stick.

Die ersten beiden Varianten können Sie sich nun auf den folgenden Seiten näher anschauen.

6.1 Veröffentlichen per E-Mail

Sie wollen Ihr öffentliches Zertifikat Ihrem Korrespondenzpartner bekannt machen? Schicken Sie ihm doch einfach ihr exportiertes öffentliches Zertifikat per E-Mail. Wie das genau funktioniert, erfahren Sie in diesem Abschnitt.

Üben Sie jetzt diesen Vorgang einmal mit Ihrem öffentlichen OpenPGP-Zertifikat! Adele soll Ihnen dabei behilflich sein. **Achtung:** Die folgenden Übungen gelten nur für OpenPGP! Anmerkungen zum Veröffentlichen von öffentlichen X.509-Zertifikaten finden Sie auf Seite 49.

Adele ist ein sehr netter E-Mail-Roboter, mit dem Sie zwanglos korrespondieren können. Weil man gewöhnlich mit einer klugen und netten jungen Dame lieber korrespondiert als mit einem Stück Software (was Adele in Wirklichkeit natürlich ist), können Sie sich Adele so vorstellen:

Schicken Sie zunächst Adele Ihr öffentliches OpenPGP-Zertifikat. Mit Hilfe des öffentlichen Schlüssel aus diesem Zertifikat sendet Ihnen Adele eine verschlüsselte E-Mail an Sie zurück.

Diese Antwort von Adele entschlüsseln Sie mit Ihrem eigenen geheimen Schlüssel. Damit Sie wiederum Adele verschlüsselt antworten können, legt Adele ihr eigenes öffentliches Zertifikat bei.

Adele verhält sich also genau wie ein richtiger Korrespondenzpartner. Allerdings sind Adeles E-Mails leider bei weitem nicht so interessant wie die Ihrer echten Korrespondenzpartner. Andererseits können Sie mit Adele so oft üben, wie Sie wollen – was Ihnen ein menschlicher Adressat wahrscheinlich ziemlich übel nehmen würde.

Exportieren Sie also nun Ihr öffentliches OpenPGP-Zertifikat und senden dieses per E-Mail an Adele. Wie das geht, erfahren Sie auf den nächsten Seiten.

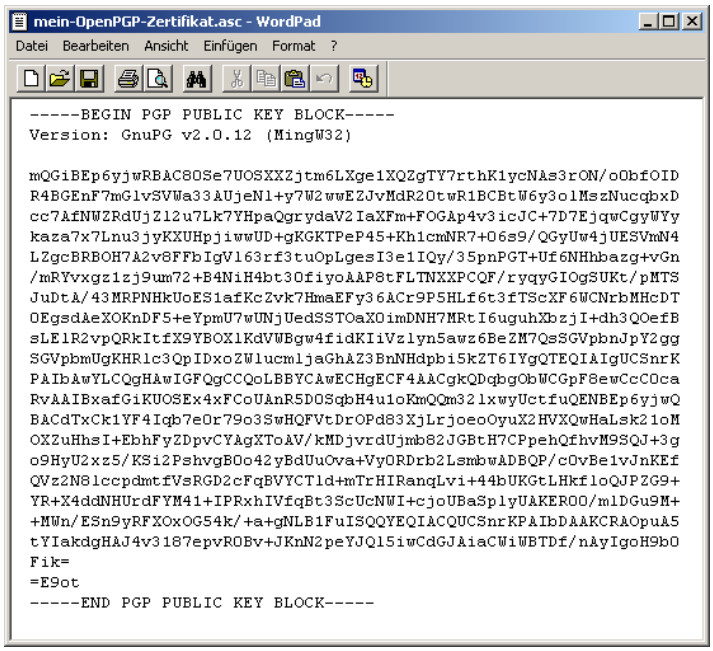
Exportieren Ihres öffentlichen OpenPGP-Zertifikats

Selektieren Sie in Kleopatra das zu exportierende öffentliche Zertifikat (durch Klicken auf die entsprechende Zeile in der Liste der Zertifikate) und klicken Sie dann auf *Datei→Zertifikate exportieren...* im Menü. Wählen Sie einen geeigneten Dateiordner auf Ihrem PC aus und speichern Sie das öffentliche Zertifikat im Dateityp `.asc` ab, z.B.: `mein-OpenPGP-Zertifikat.asc`. (Die beiden anderen zur Auswahl stehenden Dateitypen, `.gpg` oder `.pgp`, speichern Ihr Zertifikat im Binärformat. D.h. sie sind, anders als eine `.asc`-Datei, nicht im Texteditor lesbar.)

Wichtig: Achten Sie beim Auswählen des Menüpunktes darauf, dass Sie auch wirklich nur Ihr öffentliches Zertifikat exportieren – und *nicht* aus Versehen das Zertifikat Ihres kompletten Schlüsselpaars mit zugehörigem geheimen Schlüssel.

Sehen Sie sich zur Kontrolle diese Datei an. Nutzen Sie dazu den Windows Explorer und wählen denselben Order aus, den Sie beim Exportieren angegeben haben.

Öffnen Sie die exportierte Zertifikats-Datei mit einem Texteditor, z.B. mit WordPad. Sie sehen Ihr öffentliches OpenPGP-Zertifikat im Texteditor so, wie es wirklich aussieht – ein ziemlich wirrer Text- und Zahlenblock:



Variante 1: Öffentliches OpenPGP-Zertifikat als E-Mail-Text versenden

Die hier zuerst gezeigte Möglichkeit funktioniert immer, selbst wenn Sie – z.B. bei manchen E-Mail-Services im Web – keine Dateien anhängen können. Zudem bekommen Sie so Ihr öffentliches Zertifikat zum ersten Mal zu Gesicht und wissen, was sich dahinter verbirgt und woraus das Zertifikat eigentlich besteht.

Markieren Sie nun im Texteditor das gesamte öffentliche Zertifikat von

```
---BEGIN PGP PUBLIC KEY BLOCK---
```
bis
```
---END PGP PUBLIC KEY BLOCK---
```

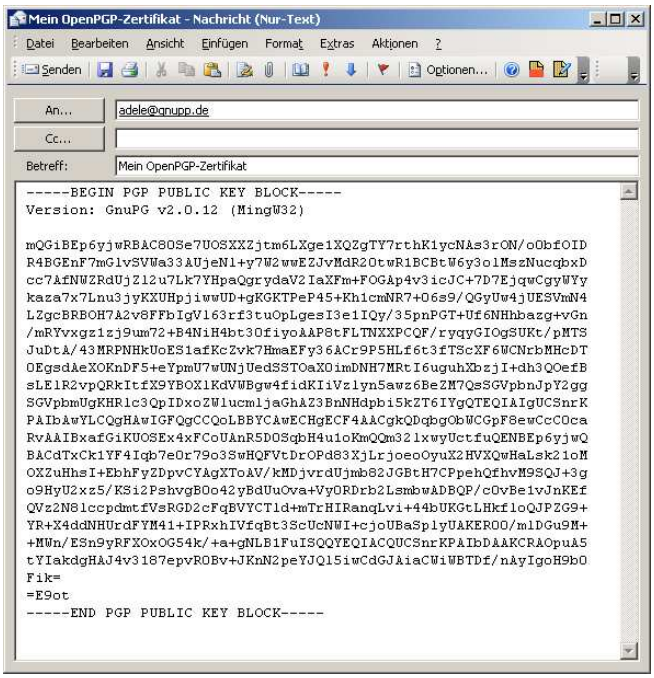
und **kopieren** Sie es mit dem Menübefehl oder mit dem Tastaturkürzel `Strg+C`. Damit haben Sie das Zertifikat in den Speicher Ihres Rechners (bei Windows Zwischenablage genannt) kopiert.

Nun starten Sie Ihr E-Mail-Programm – es spielt keine Rolle, welches Sie benutzen – und fügen Ihr öffentliches Zertifikat in eine leere E-Mail ein. Der Tastaturbefehl zum Einfügen („Paste“) lautet bei Windows `Strg+V`. Es ist sinnvoll vorher das E-Mail-Programm so zu konfigurieren, dass reine Textnachrichten gesendet werden und keine HTML-formatierte Nachrichten (vgl. Abschnitt 10.3 und Anhang B).

Diesen Vorgang – Kopieren und Einfügen – kennen Sie vielleicht als „Copy & Paste“.

Adressieren Sie nun diese E-Mail an `adele@gnupp.de` und schreiben in die Betreffzeile z.B. *Mein öffentliches OpenPGP-Zertifikat*.

So etwa sollte Ihre E-Mail nun aussehen:



Schicken Sie die E-Mail an Adele ab.

Nur zur Vorsicht: Natürlich sollten Ihre E-Mails nicht `heinrichh@gpg4win.de` oder ein andere Beispieladresse als Absender haben, sondern Ihre *eigene* E-Mail-Adresse. Denn sonst werden Sie nie Antwort von Adele bekommen. . .

Variante 2: Öffentliches OpenPGP-Zertifikat als E-Mail-Anhang versenden

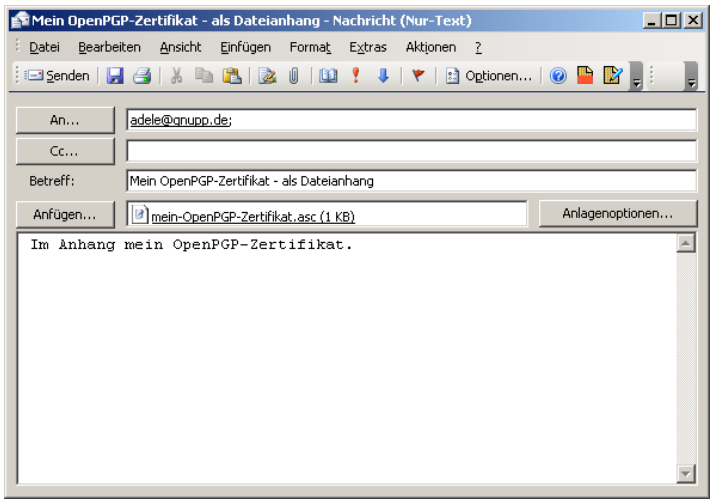
Alternativ zu Variante 1 können Sie natürlich Ihr exportiertes öffentliches OpenPGP-Zertifikat auch direkt als **E-Mail-Dateianhang** versenden. Das ist oftmals das einfachere und gebräuchlichere Verfahren. Sie haben oben die „Copy & Paste“-Methode zuerst kennengelernt, weil sie transparenter und leichter nachzuvollziehen ist.

Schreiben Sie Adele nun noch einmal eine neue E-Mail – diesmal mit der Zertifikatsdatei im Anhang:

Fügen Sie die oben exportierte Zertifikatsdatei als Anhang zu Ihrer neuen E-Mail hinzu – genauso wie Sie es mit jeder anderen Datei auch machen (z.B. durch Ziehen der Datei in das leere E-Mail-Fenster). Ergänzen Sie den Empfänger (`adele@gnupp.de`) und einen Betreff, z.B.: *Mein öffentliches OpenPGP-Zertifikat - als Dateianhang*.

Selbstverständlich dürfen Sie auch noch ein paar erklärende Sätze dazuschreiben. Adele braucht diese Erklärung jedoch nicht, denn sie ist zu nichts anderem als zu diesem Übungszweck programmiert worden.

Ihre fertige E-Mail sollte dann etwa so aussehen:



Senden Sie nun die E-Mail mit Anhang an Adele ab.

Kurz zusammengefasst

Sie haben Ihr öffentliches OpenPGP-Zertifikat in Kleopatra in eine Datei exportiert. Anschließend haben Sie einmal den Inhalt der Datei direkt in eine E-Mail kopiert und einmal die komplette Datei als E-Mail-Anhang beigefügt. Beide E-Mails haben Sie an einen Korrespondenzpartner (in Ihrem Fall Adele) geschickt.

Genauso gehen Sie vor, wenn Sie Ihr öffentliches Zertifikat an eine echte E-Mail-Adresse senden. Sie entscheiden sich dabei natürlich für eine der beiden oben vorgestellten Varianten – in der Regel sollten Sie öffentliche Zertifikate als Dateianhang versenden. Dies ist für Sie und Ihren Empfänger das Einfachste. Und es hat den Vorteil, dass Ihr Empfänger Ihre Zertifikatsdatei direkt (ohne Umwege) in seine Zertifikatsverwaltung (z.B. Kleopatra) importieren kann.

S/MIME

Nachdem Sie gelernt haben, wie Sie Ihr öffentliches OpenPGP-Zertifikat per E-Mail veröffentlichen, wird Sie sicher interessieren wie das Gleiche für öffentliche **X.509-Zertifikate** funktioniert (vgl. auch Kapitel 3).

Die Antwort lautet: Sie können es so wie bei OpenPGP machen. Sie exportieren Ihr öffentliches X.509-Zertifikat in Kleopatra, speichern dieses z.B. im Dateiformat `.pem` ab und versenden die Datei als E-Mail-Anhang. Aber konkret ist das bei S/MIME unnötig. Es genügt, wenn Sie Ihrem Korrespondenzpartner eine signierte S/MIME-E-Mail senden. Ihr öffentliches X.509-Zertifikat ist in dieser Signatur enthalten und kann von dem Empfänger in die Zertifikatsverwaltung importiert werden.

Der einzige Unterschied zum oben beschriebenen OpenPGP-Vorgehen: Sie können Adele nicht benutzen! **Adele unterstützt nur OpenPGP!** Zum Üben sollten Sie sich also einen anderen Korrespondenzpartner aussuchen oder Sie schreiben testweise an sich selber.

Beim Exportieren Ihres öffentlichen X.509-Zertifikats haben Sie die Wahl, ob Sie die vollständige öffentliche Zertifikatskette (in der Regel: Wurzelzertifikat – CA-Zertifikat – Persönliches Zertifikat) oder nur Ihr öffentliches Zertifikat in eine Datei abspeichern wollen. Ersteres ist empfehlenswert, denn Ihrem Korrespondenzpartner fehlen möglicherweise Teile der Kette, die er sonst zusammensuchen müsste. Klicken Sie dazu in Kleopatra alle Elemente einer Zertifikatskette mit gedrückter Shift-Taste an und exportieren Sie diese markierten Komponenten.

Hatte Ihr Korrespondenzpartner das Wurzelzertifikat noch nicht, so muss er diesem Wurzelzertifikat das Vertrauen aussprechen bzw. durch einen Administrator aussprechen lassen, um letztlich auch Ihnen zu vertrauen. Ist das bereits vorher geschehen (z.B. weil sie beide zu der selben „Wurzel“ gehören), dann besteht diese Vertrauensstellung bereits, und ist wirksam unmittelbar mit der Verfügbarkeit der Kette.

6.2 Veröffentlichen per OpenPGP-Zertifikatsserver

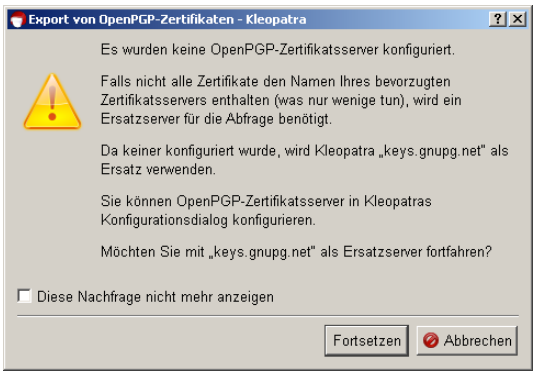
Wichtig: Die Verbreitung Ihres öffentlichen Zertifikats auf einem OpenPGP-Zertifikatsserver ist nur für OpenPGP-Zertifikate möglich!

Die Publizierung Ihres öffentlichen OpenPGP-Zertifikats auf einem öffentlichen Zertifikatsserver bietet sich eigentlich immer an, selbst wenn Sie nur mit wenigen Partnern verschlüsselte E-Mails austauschen. Ihr öffentliches Zertifikat ist dann für jedermann zugänglich auf einem Server im Internet verfügbar. Sie ersparen sich dadurch die Versendung Ihres Zertifikats per E-Mail an jeden Ihrer Korrespondenzpartner.

Vorsicht: Die Veröffentlichung Ihrer E-Mail-Adresse auf einem Zertifikatsserver birgt leider das Risiko, dass Ihnen auch ungebetene Personen E-Mails schreiben können und die SPAM-Menge für Ihre E-Mail-Adresse dadurch zunehmen kann. Sie sollten daher im zweiten Fall einen ausreichenden SPAM-Schutz nutzen. Falls Sie keinen wirksamen Spamfilter benutzen, sollten Sie u.U. von der Veröffentlichung Ihres öffentlichen Zertifikats auf einem Zertifikatsserver absehen.

Und so geht's: Wählen Sie Ihr öffentliches OpenPGP-Zertifikat in Kleopatra aus und klicken im Menü auf *Datei→Zertifikate nach Server exportieren...*.

Sofern Sie noch keinen Zertifikatsserver definiert haben, bekommen Sie eine Warnmeldung:



Wie Sie an der Meldung erkennen können, ist der öffentliche OpenPGP-Zertifikatsserver `keys.gnupg.net` bereits voreingestellt. Klicken Sie auf [*Fortsetzen*], um Ihr ausgewähltes öffentliches Zertifikat an diesen Server zu schicken. Von dort aus wird Ihr öffentliches Zertifikat an alle, weltweit verbundenen Zertifikatsserver weitergereicht. Jedermann kann Ihr öffentliches Zertifikat dann von einen dieser OpenPGP-Zertifikatsserver herunterladen und dazu benutzen, Ihnen eine sichere E-Mail zu schreiben.

Wenn Sie den Ablauf nur testen, dann schicken Sie das Übungszertifikat bitte nicht ab, indem Sie im obigen Dialog auf [*Abbrechen*] klicken. Er ist wertlos und kann nicht mehr vom Zertifikatsserver entfernt werden. Sie glauben nicht, wieviele Testkeys mit Namen wie „Julius Caesar", „Helmut Kohl" oder „Bill Clinton" dort schon seit Jahren herumliegen. . .

Kurz zusammengefasst

Sie wissen nun, wie Sie Ihr öffentliches OpenPGP-Zertifikat auf einen OpenPGP-Zertifikatsserver im Internet veröffentlichen.

Wie Sie das öffentliche OpenPGP-Zertifikat eines Korrespondenzpartners auf Zertifikatsservern suchen und importieren, erfahren Sie im Kapitel 15. Sie können dieses Kapitel jetzt lesen oder später, wenn Sie diese Funktion benötigen.

S/MIME

Die Verbreitung von öffentlichen X.509-Zertifikaten erfolgt in einigen Fällen durch die Beglaubigungsinstanz. Das passiert typischerweise über X.509-Zertifikatsserver, die per LDAP erreichbar sind. Im Unterschied zu den OpenPGP-Zertifikatsservern synchronisieren sich die X.509-Zertifikatsserver jedoch nicht weltweit untereinander.

7 Sie entschlüsseln eine E-Mail

Sie bekommen verschlüsselte Nachrichten Ihrer Korrespondenzpartner und wollen diese nun entschlüsseln? Alles was Sie dazu brauchen ist Gpg4win, das Zertifikat Ihres Schlüsselpaars und natürlich ganz wichtig: Ihre Passphrase.

In diesem Kapitel bekommen Sie Schritt für Schritt erklärt, wie Sie Ihre E-Mails in Microsoft Outlook mit der Gpg4win-Programmkomponente GpgOL entschlüsseln.

Üben Sie jetzt diesen Vorgang einmal mit Adele und Ihrem öffentlichen OpenPGP-Zertifikat!

Achtung: Die folgenden Übungen gelten **nur für OpenPGP**! Anmerkungen zur Entschlüsselung von S/MIME-E-Mails finden Sie am Ende dieses Kapitels auf Seite 55.

Im Abschnitt 6.1 haben Sie Adele Ihr öffentliches OpenPGP-Zertifikat geschickt. Mit Hilfe dieses Zertifikats verschlüsselt Adele nun eine E-Mail und sendet die Nachricht an Sie zurück. Nach kurzer Zeit sollten Sie Adeles Antwort erhalten.

Nachricht mit MS Outlook und GpgOL entschlüsseln

Für die meisten E-Mail-Programme gibt es spezielle Programmerweiterungen (engl. „plugins“), mit denen die Ver- und Entschlüsselung direkt im jeweiligen E-Mail-Programm erledigt werden kann – **GpgOL** ist eine solche Programmerweiterung für MS Outlook, dass in diesem Abschnitt benutzt wird, um die E-Mail von Adele zu entschlüsseln.

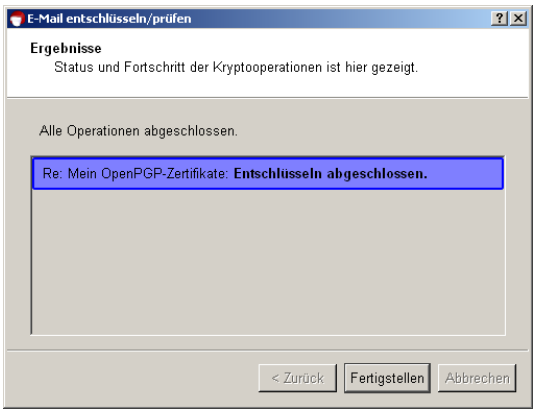
Hinweise zu weiteren Software-Lösungen finden Sie im Anhang C. Sie können diesen Abschnitt jetzt lesen oder später, wenn Sie diese Funktion benötigen.

Starten Sie MS Outlook und öffnen Sie die Antwort-E-Mail von Adele.

Kleopatra haben Sie bisher nur als Zertifikatsverwaltung kennengelernt. Das Programm leistet aber weitaus mehr: Es kann die eigentliche Verschlüsselungs-Software GnuPG steuern und damit nicht nur Ihre Zertifikate verwalten, sondern auch sämtliche kryptographischen Aufgaben (eben mit Hilfe von GnuPG) erledigen. Kleopatra sorgt für die graphische Benutzeroberfläche, also die Dialoge, die Sie als Benutzer sehen während Sie eine E-Mail ver- oder entschlüsseln. Das heißt auch, dass Sie immer die gleichen Dialog sehen, egal ob Sie mit Outlook, einem anderen E-Mail-Programm oder auch mit dem Windows Explorer etwas verschlüsseln.

Kleopatra bearbeitet also die verschlüsselte E-Mail von Adele. Diese E-Mail hat Adele mit *Ihrem* öffentlichen OpenPGP-Schlüssel verschlüsselt.

Um die Nachricht zu entschlüsseln, fragt Kleopatra Sie nun nach Ihrer (Ihren privaten Schlüssel schützenden) Passphrase. Geben Sie diese in den aufkommenden Dialog ein. Sofern Ihre Eingabe korrekt war, erhalten Sie einen Statusdialog (siehe nachfolgende Abbildung). Mit [*Details anzeigen*] können Sie sich weitere Informationen der E-Mail-Überprüfung anzeigen lassen.



Die Entschlüsselung war erfolgreich! Schließen Sie den Dialog, um die entschlüsselte E-Mail zu lesen.

Möchten Sie den Prüfdialog nach dem Lesen der E-Mail noch einmal manuell aufrufen, so klicken Sie im Menü der geöffneten E-Mail auf *Extras→GpgOL Entschlüsseln / Prüfen*.

Doch nun wollen Sie sicher das Ergebnis, die entschlüsselte Nachricht, endlich einmal sehen...

Die entschlüsselte Nachricht

Die entschlüsselte Antwort von Adele sieht in etwa so aus[1]:

```
Hallo Heinrich Heine,

hier ist die verschlüsselte Antwort auf Ihre E-Mail.

Ihr öffentlicher Schlüssel mit der Schlüssel-ID
0EA6E039B5821A91 und der Bezeichnung
‘Heinrich Heine <heinrichh@gpg4win.de>’
wurde von mir empfangen.

Anbei der öffentliche Schlüssel von adele@gnupp.de,
dem freundlichen E-Mail-Roboter.

Viele Grüße,
adele@gnupp.de
```

Der Textblock, der darauf folgt, ist das öffentliche Zertifikat von Adele.

Im nächsten Kapitel werden Sie dieses Zertifikat importieren und zu Ihrer Zertifikatsverwaltung hinzufügen. Importierte öffentliche Zertifikate können Sie jederzeit zum Verschlüsseln von Nachrichten an Ihren Korrespondenzpartner benutzen oder dessen signierte E-Mails prüfen.

[1] Abhängig von der Softwareversion von Adele kann dies auch etwas unterschiedlich aussehen.

Kurz zusammengefasst

1. Sie haben eine verschlüsselte E-Mail mit Ihrem geheimen Schlüssel entschlüsselt.
2. Ihr Korrespondenzpartner hat sein eigenes öffentliches Zertifikat beigelegt, damit Sie ihm verschlüsselt antworten können.

S/MIME

Nachdem Sie gelernt haben, wie Sie E-Mails mit Ihrem geheimen OpenPGP-Schlüssel entschlüsseln, werden Sie nun noch erfahren, wie Sie verschlüsselte **S/MIME**-E-Mails entschlüsseln.

Die Antwort lautet auch hier: Genauso wie bei OpenPGP! Der Unterschied zu OpenPGP ist lediglich, dass S/MIME *nicht* von Adele unterstützt wird und somit die obige Übung nur für OpenPGP gilt.

Zum Entschlüsseln einer S/MIME-verschlüsselten E-Mail öffnen Sie die Nachricht in Outlook und geben im Pinentry-Dialog Ihre Passphrase ein. Sie bekommen einen ähnlichen Statusdialog wie bei OpenPGP. Nach dem Schließen dieses Dialogs sehen Sie die entschlüsselte S/MIME E-Mail.

8 Sie importieren ein öffentliches Zertifikat

Ihr Korrespondenzpartner muss nicht jedes Mal sein öffentliches Zertifikat mitschicken, wenn er Ihnen signiert schreibt. Sie bewahren sein öffentliches Zertifikat einfach in Ihrer Zertifikatsverwaltung (z.B. Kleopatra) auf.

Öffentliches Zertifikat abspeichern

Bevor Sie ein öffentliches Zertifikat in Kleopatra importieren, müssen Sie es in einer Datei abspeichern. Abhängig davon, ob Sie das Zertifikat als E-Mail-Dateianhang oder als Textblock innerhalb Ihrer E-Mail bekommen haben, gehen Sie wie folgt vor:

- Liegt das öffentliche Zertifikat einer E-Mail als **Dateianhang** bei, speichern Sie es (wie Sie es in Ihrem E-Mail-Programm gewohnt sind) auf einem Ort Ihrer Festplatte ab.
- Sollte das öffentliche Zertifikat als **Textblock** innerhalb Ihrer E-Mail übermittelt worden sein, so müssen Sie zunächst das vollständige Zertifikat markieren:

  Bei (öffentlichen) OpenPGP-Zertifikaten markieren Sie den Bereich von

  `---BEGIN PGP PUBLIC KEY BLOCK---`
  bis
  `---END PGP PUBLIC KEY BLOCK---`

  so wie Sie es im Abschnitt 6.1 schon getan haben.

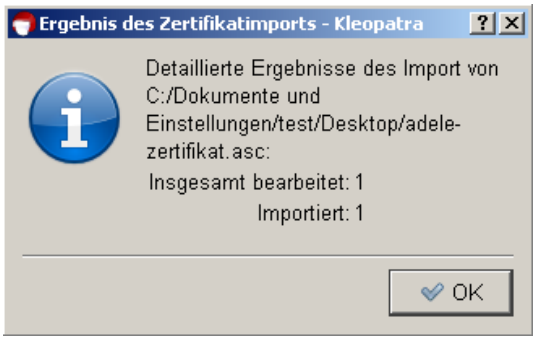
  Setzen Sie den markierten Abschnitt per Copy & Paste in einen Texteditor ein und speichern Sie das öffentliche Zertifikat ab. Als Dateiendung sollten Sie für OpenPGP-Zertifikate `.asc` oder `.gpg` und für X.509-Zertifikate `.pem` oder `.der` wählen.

Öffentliches Zertifikat in Kleopatra importieren

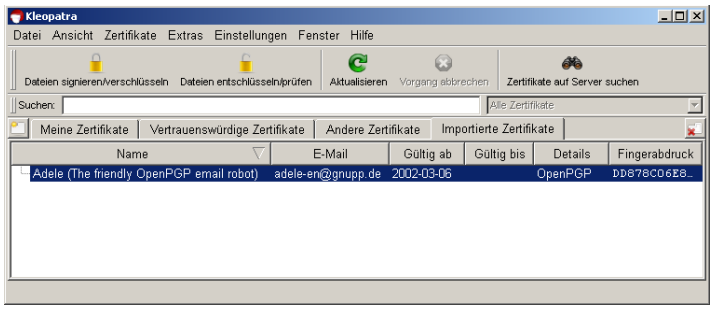
Ob Sie nun das öffentliche Zertifikat als E-Mail-Anhang oder als Textblock abgespeichert haben, ist egal: In beiden Fällen importieren Sie dieses abgespeicherte Zertifikat in Ihre Zertifikatsverwaltung Kleopatra.

Starten Sie dafür Kleopatra, sofern das Programm noch nicht läuft.

Klicken Sie im Menü auf *Datei→Zertifikat importieren...*, suchen das eben abgespeicherte öffentliche Zertifikat aus und importieren es. Sie erhalten einen Informations-Dialog mit dem Ergebnis des Importvorgangs:



Das erfolgreich importierte, öffentliche Zertifikat wird nun in Kleopatra angezeigt – und zwar unter einem separatem Reiter *Importierte Zertifikate* von <*Pfad-zur-Zertifikatsdatei*>“:



Dieser Reiter dient zur Kontrolle, weil einer Datei durchaus auch mehr als nur ein Zertifikat enthalten kann. Schließen Sie diesen Reiter über das Menü *Fenster→Reiter schließen* (oder über die „Reiter-schließen“ Schaltfläche am rechten Fensterrand.

Wechseln Sie auf den Tab „Andere Zertifikate“. Hier sollten Sie nun das von Ihnen importierte öffentliche Zertifikat sehen.

Damit haben Sie ein fremdes Zertifikat – in diesem Beispiel das öffentliche OpenPGP-Zertifikat von Adele – in Ihre Zertifikatsverwaltung importiert. Sie können dieses Zertifikat jederzeit benutzen, um verschlüsselte Nachrichten an den Besitzer dieses Zertifikats zu senden und dessen Signaturen zu prüfen.

Sobald Sie E-Mail-Verschlüsselung häufiger und mit vielen Korrespondenzpartnern betreiben, werden Sie aus Gründen des Komforts die Zertifikate über global erreichbare Zertifikatsserver suchen und importieren wollen. Wie das geht, können Sie im Kapitel 15 nachlesen.

Bevor Sie weitermachen, eine wichtige Frage:

Woher wissen Sie eigentlich, dass das öffentliche OpenPGP-Zertifikat wirklich von Adele stammt? Man kann E-Mails auch unter falschem Namen versenden – die Absenderangabe besagt eigentlich gar nichts.

Wie können Sie also sichergehen, dass ein öffentliches Zertifikat auch wirklich seinem Absender gehört?

Diese Kernfrage der Zertifikatsprüfung wird im Kapitel 16 erläutert. Lesen Sie bitte jetzt dort weiter, bevor Sie danach an dieser Stelle fortfahren.

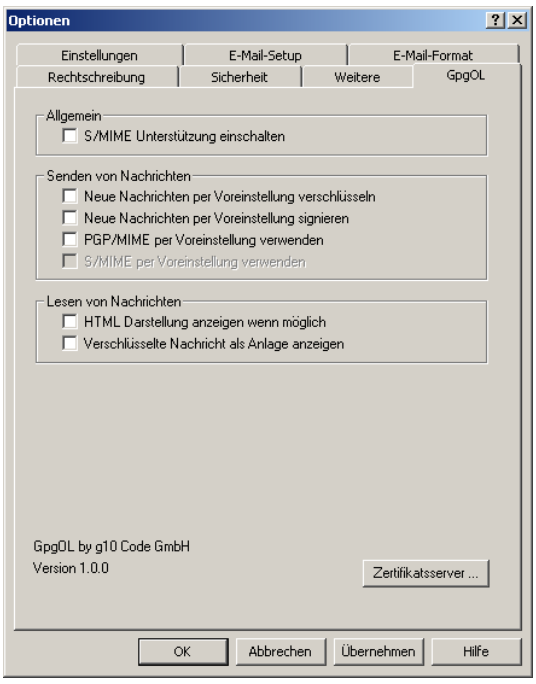
9 Sie verschlüsseln eine E-Mail

Jetzt wird es noch einmal spannend: Sie versenden eine verschlüsselte E-Mail!

In diesem Beispiel brauchen Sie dazu Outlook (das geht auch mit anderen E-Mail-Klienten, die Kryptographie unterstützen), Kleopatra und natürlich ein öffentliches Zertifikat Ihres Korrespondenzpartners.

Hinweis nur für OpenPGP:
Sie können zum Üben dieses Vorgangs mit OpenPGP wieder Adele nutzen. S/MIME wird von Adele nicht unterstützt! Adressieren Sie dazu Ihre zu verschlüsselnde E-Mail an `adele@gnupp.de`. Der Inhalt der Nachricht ist beliebig – Adele kann nicht wirklich lesen...

Hinweis nur für S/MIME:
Nach der Installation von Gpg4win ist die S/MIME-Funktionalität in GpgOL deaktiviert. Wenn Sie S/MIME (mit GnuPG) nutzen möchten, müssen Sie zuvor wie in dem nachfolgend dargestellten GpgOL-Optionsdialog unter *Extras→Optionen→GpgOL* die Option *S/MIME Unterstützung einschalten* aktivieren:



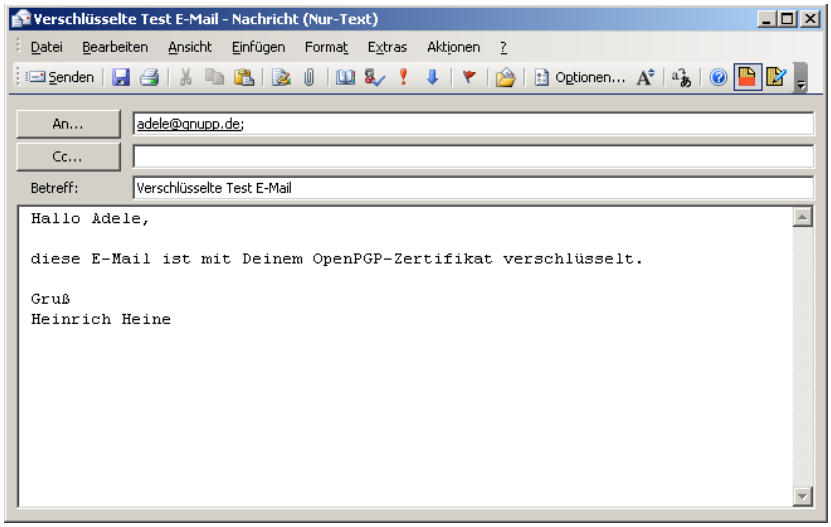
Lesen Sie sich die angezeigten Informationen sorgfältig durch, vor allem dann wenn Sie schon vorher mit S/MIME unter einem anderen Kryptographie-Produkt gearbeitet haben.

Nachricht verschlüsselt versenden

Verfassen Sie zunächst in Outlook eine neue E-Mail und adressieren Sie diese an Ihren Korrespondenzpartner.

Jetzt fehlt nur noch die Angabe, dass Sie Ihre Nachricht auch wirklich verschlüsselt versenden wollen: Wählen Sie im Menü des Nachrichtenfensters den Punkt *Extras→Nachricht verschlüsseln*. Die Schaltfläche mit dem Schloss-Icon in der Symbolleiste ist aktiviert (Sie können auch gleich direkt auf diese Schaltfläche klicken).

Ihre Outlook-Nachrichtenfenster sollte nun etwa so aussehen:



Um die Verschlüsselungsoption wieder zu deaktivieren genügt ein erneuter Klick auf die o.g. Schaltfläche.

Klicken Sie nun auf [*Senden*].

Hinweis:
Im Normalfall erkennt Gpg4win automatisch in welchem Protokoll (OpenPGP oder S/MIME) das passende öffentliche Zertifikat Ihres Korrespondenzpartners vorliegt.

Bei Bedarf können Sie das Protokoll explizit für diese E-Mail über das Menü *Extras→GnuPG Protokoll* im geöffneten Outlook-Nachrichtenfensters auf *PGP/MIME*, *S/MIME* oder *automatisch* (Standardeinstellung) setzen.

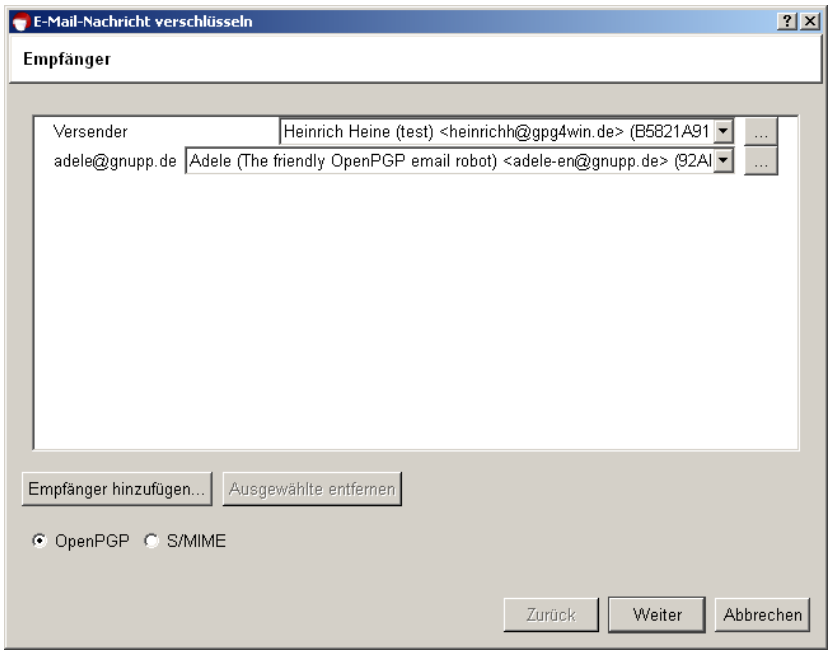
Wenn Sie sich unsicher sind, belassen Sie es bei der Voreinstellung *automatisch*. Sie haben später im Verschlüsselungsprozess noch die Möglichkeit zwischen PGP/MIME und S/MIME zu wählen.

Haben Sie ein bevorzugtes GnuPG-Protokoll?
Dann können Sie unter den GpgOL-Optionen (*Extras→Optionen→GpgOL*) PGP/MIME oder S/MIME für alle signierte und verschlüsselte Nachrichten als Voreinstellung definieren.

Zertifikatsauswahl

Daraufhin öffnet Kleopatra ein Fenster, in dem Sie das öffentliche Zertifikat des Empfängers bestätigen müssen. Kleopatra wählt – abhängig von der Empfänger-E-Mail-Adresse – in der Regel das passende öffentliche Zertifikat aus Ihrer Zertifikatsverwaltung automatisch aus.

Sollte Kleopatra das öffentliche Zertifikat Ihres Korrespondenzpartners nicht finden, haben Sie es vermutlich noch nicht in Ihrer Zertifikatsverwaltung importiert (vgl. Kapitel 8).

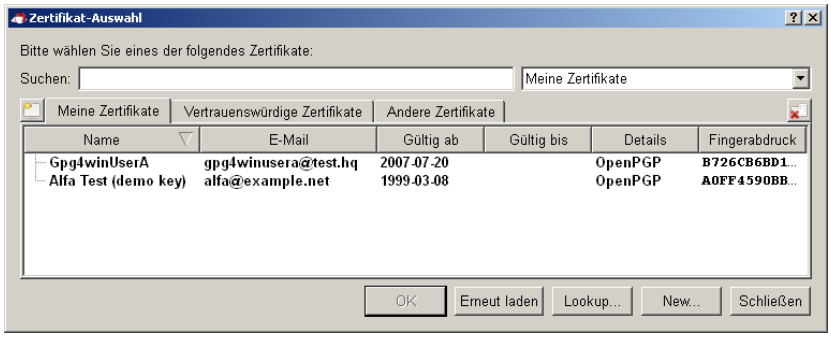


Im Normalfall können Sie dieses vorausgewählte öffentliche Zertifikat mit [*Weiter*] bestätigen.

Sollte Kleopatra im Dialog auf der vorherigen Seite **nicht** das richtige öffentliche Zertifikat ausgewählt haben (z.B. weil zu der Empfänger-E-Mail-Adresse mehrere öffentliche Zertifikate existieren), dann drücken Sie auf die [...]-Schaltfläche neben der Auswahlliste.

Sie können hier auch bewusst ein anderes öffentliches Zertifikat (eines das nicht zu der E-Mail-Adresse passt) auswählen. Das macht praktisch aber wenig Sinn und kann Ihnen (und Ihrem Empfänger) eher Probleme bereiten...

Sie bekommen einen Kleopatra-Dialog mit einer Liste aller öffentlichen Zertifikate des gewählten Zertifikatstyps angezeigt, die in Ihrer Zertifikatsverwaltung existieren (also der öffentlichen Zertifikate, die von Ihnen bis dahin importiert wurden). Exemplarisch sehen Sie hier eine Auswahl von verfügbaren öffentlichen OpenPGP-Zertifikaten:



Wählen Sie das korrekte öffentliche Zertifikat Ihres Korrespondenzpartners aus, denn damit muss Ihre Nachricht schließlich verschlüsselt werden.

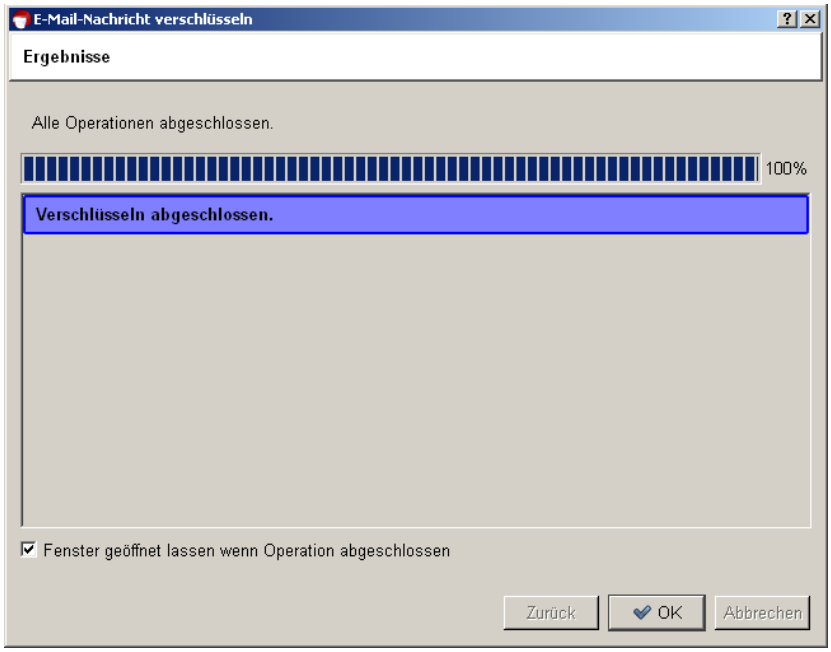
Sie erinnern sich an den Grundsatz:

> **Wenn Sie an jemanden verschlüsselte E-Mails schicken wollen, müssen Sie dessen öffentliches Zertifikat haben und zum Verschlüsseln benutzen.**

Klicken Sie auf [*Weiter*]. Ihre Nachricht wird nun verschlüsselt.

Verschlüsselung abschließen

Wurde Ihre Nachricht erfolgreich verschlüsselt und versandt, erhalten Sie eine Meldung, die Ihnen dies bestätigt:



Herzlichen Glückwunsch! Sie haben Ihre erste E-Mail verschlüsselt!

10 Sie signieren eine E-Mail

Sie haben in Kapitel 16 gelesen, wie Sie sich von der Echtheit eines öffentlichen OpenPGP-Zertifikats überzeugen und es dann mit Ihrem eigenen geheimen OpenPGP-Schlüssel signieren können.

Dieses Kapitel beschäftigt sich damit, wie Sie nicht nur ein Zertifikat, sondern auch eine komplette **E-Mail signieren** können. Das bedeutet, dass Sie die E-Mail mit einer elektronischen Signatur (einer Art elektronischem Siegel) versehen.

Der Text ist dann zwar noch für jeden lesbar, aber Ihr Empfänger kann feststellen, ob die E-Mail unterwegs manipuliert oder verändert wurde.

Die Signatur garantiert Ihrem Empfänger, dass die Nachricht tatsächlich von Ihnen stammt. Und: wenn Sie mit jemandem korrespondieren, dessen öffentliches Zertifikat Sie nicht haben (aus welchem Grund auch immer), können Sie so die Nachricht wenigstens mit Ihrem eigenen privaten Schlüssel „versiegeln“.

Achtung: Verwechseln Sie diese elektronische Signatur nicht mit der E-Mail-„Signatur“, die man unter eine E-Mail setzt und die z.B. Ihre Telefonnummer, Ihre Adresse und Ihre Webseite enthalten. Während diese E-Mail-Signaturen einfach nur als eine Art Visitenkarte fungieren, schützt die elektronische Signatur Ihre E-Mail vor Manipulationen und bestätigt den Absender.

Übrigens ist die elektronische Signatur auch nicht mit der qualifizierten digitalen Signatur gleichzusetzen, wie sie im Signaturgesetz vom 22. Mai 2001 in Kraft getreten ist. Für die private oder berufliche E-Mail-Kommunikation erfüllt sie allerdings genau denselben Zweck.

10.1 Signieren mit GpgOL

Tatsächlich ist die Signierung einer E-Mail noch einfacher als die Verschlüsselung (vgl. Kapitel 9). Nachdem Sie eine neue E-Mail verfasst haben, gehen Sie – analog zur Verschlüsselung – folgende Schritte durch:

- Nachricht signiert versenden
- Zertifikatsauswahl
- Signierung abschließen

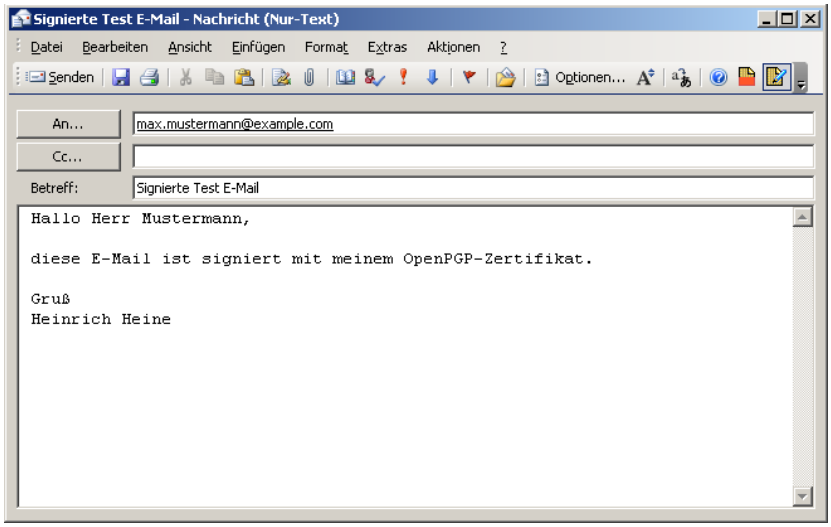
Auf den nächsten Seiten werden diese Schritte im Detail beschrieben.

Nachricht signiert versenden

Verfassen Sie zunächst in Outlook eine neue E-Mail und adressieren Sie diese an Ihren Korrespondenzpartner.

Bevor Sie Ihre Nachricht abschicken, geben Sie noch an, dass Ihre Nachricht signiert versendet werden soll: Dazu aktivieren Sie den Menüeintrag *Extras→Nachricht signieren*. Die Schaltfläche mit dem signierenden Stift wird aktiviert.

Ihr E-Mail-Fenster sollte anschließend etwa so aussehen:



Wie beim Verschlüsseln können Sie natürlich auch die Signieroption jederzeit mit einem erneuten Klick auf die Schaltfläche wieder deaktivieren.

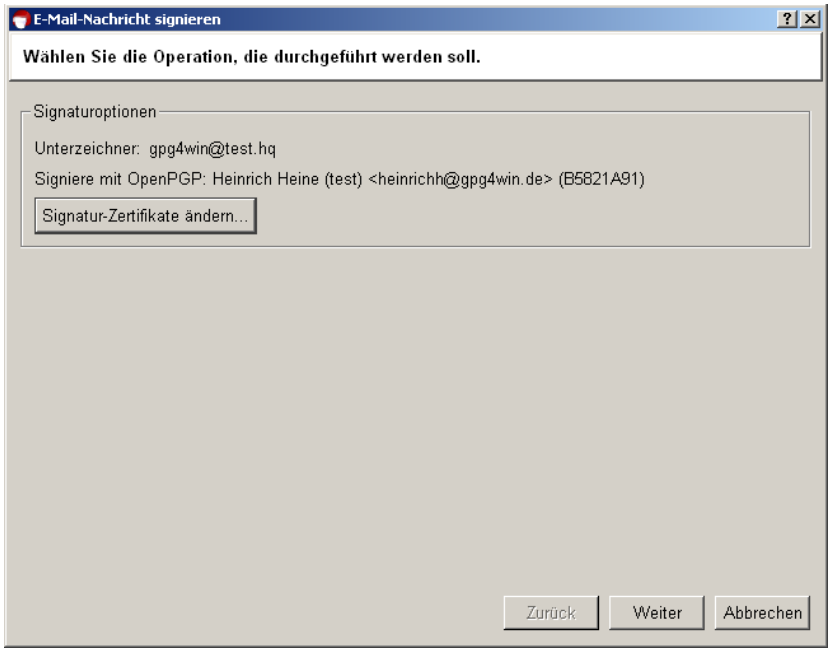
Klicken Sie nun auf [*Senden*].

Hinweis:
Genauso wie beim Verschlüsseln von E-Mails erkennt Gpg4win automatisch in welchen Protokoll (OpenPGP oder S/MIME) Ihre eignes Zertifikat (mit dem geheimen Schlüssel zum Signieren) vorliegt.

Sollen Sie das Protokoll manuell setzen wollen, nutzen Sie dazu das Menü *Extras→GnuPG Protokoll* im Outlook-Nachrichtenfenster und wählen Sie *PGP/MIME*, *S/MIME* oder *automatisch* – die Erläuterungen und Hinweise vom Verschlüsseln (siehe Seite 60) gelten auch für das Signieren.

Zertifikatsauswahl

Daraufhin öffnet Kleopatra ein Fenster, in dem – anders als beim Verschlüsseln – Ihre **eigenen** Zertifikate (also alle die Zertifikate, zu denen Ihnen jeweils ein geheimer Schlüssel vorliegt) angezeigt werden.



Denn:

> **Signieren können Sie nur mit Ihrem eigenen geheimen Schlüssel.**

Logisch, denn nur Ihr geheimer Schlüssel bestätigt Ihre Identität. Der Korrespondenzpartner kann dann mit Ihrem öffentlichen Zertifikat, das er bereits hat oder sich besorgen kann, Ihre Identität prüfen. Denn nur Ihr geheimer Schlüssel passt zu Ihrem öffentlichen Zertifikat.

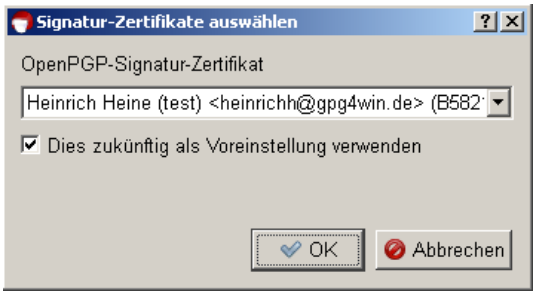
Im Normalfall können Sie im obigen Dialog Ihr vorausgewähltes Zertifikat mit [*Weiter*] bestätigen.

Beim ersten Durchlauf des Signierprozesses müssen Sie zunächst Kleopatra Ihr bevorzugtes Zertifikat zum Signieren mit OpenPGP bzw. S/MIME mitteilen. Sollte also noch kein Zertifikat ausgewählt oder nicht Ihr richtiges Zertifikat angezeigt sein – z.B. weil Sie mehrere Zertifikate besitzen – klicken Sie auf [*Signaturzertifikate ändern...*].

Sie möchten Ihr Signaturzertifikat ändern?

Dann erhalten Sie an dieser Stelle einen Auswahl-Dialog mit einer Auflistung aller Ihrer eigenen Zertifikate des vorher gewählten Protokolls, die in Ihrer Zertifikatsverwaltung existieren.

Nachfolgend ein Beispieldialog für OpenPGP:



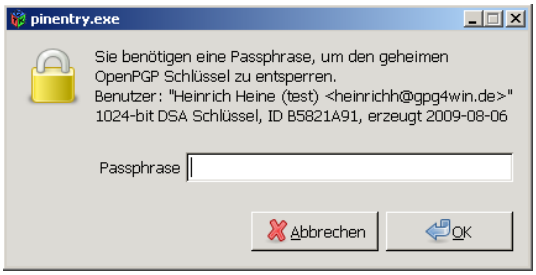
Wählen Sie Ihr korrektes Zertifikat aus, mit dem Sie Ihre Nachricht signieren wollen.

Klicken Sie anschließend auf [*OK*]. Ihr ausgewähltes Zertifikat wird in den letzten „E-Mail signieren“-Dialog übernommen.

Bestätigen Sie Ihr Zertifikat mit [*Weiter*].

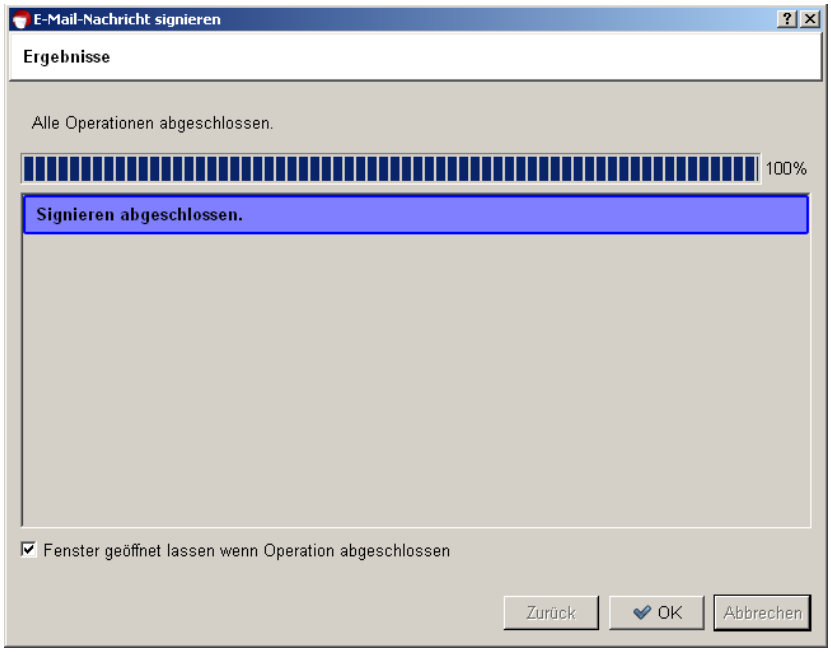
Signierung abschließen

Um die Signierung Ihrer E-Mail abzuschließen, werden Sie aufgefordert im folgenden Fenster Ihre geheime Passphrase einzugeben:



Dies ist notwendig, weil Sie mit Ihrem eignen geheimen Schlüssel signieren. Bestätigen Sie Ihre Eingabe mit [*OK*].

Ihre Nachricht wird nun signiert und versandt. Nach erfolgreicher Signierung Ihrer Nachricht, erhalten Sie folgenden Dialog:



Herzlichen Glückwunsch! Sie haben Ihre erste E-Mail signiert!

Kurz zusammengefasst

Sie haben gelernt, wie Sie eine E-Mail mit Ihrem eigenen Zertifikat (das Ihren geheimen Schüssel enthält) **signieren**.

Seit dem letzten Kapitel wissen Sie nun auch, wie Sie eine E-Mail mit dem öffentlichen Zertifikat Ihres Korrespondenzpartners **verschlüsseln**.

Damit beherrschen Sie nun die beiden wichtigsten Techniken für einen sicheren E-Mail-Versand: verschlüsseln und signieren.

Natürlich können Sie beide Techniken auch kombinieren. Entscheiden Sie ab sofort bei jeder neuen E-Mail, wie Sie Ihre Nachricht versenden wollen – je nachdem, wie wichtig und schutzbedürftig der Inhalt Ihrer E-Mail ist:

- unverschlüsselt
- verschlüsselt
- signiert
- signiert und verschlüsselt (Mehr dazu im Abschnitt 10.4)

Diese vier Kombinationen können Sie entweder mit OpenPGP oder mit S/MIME realisieren.

10.2 Signatur mit GpgOL prüfen

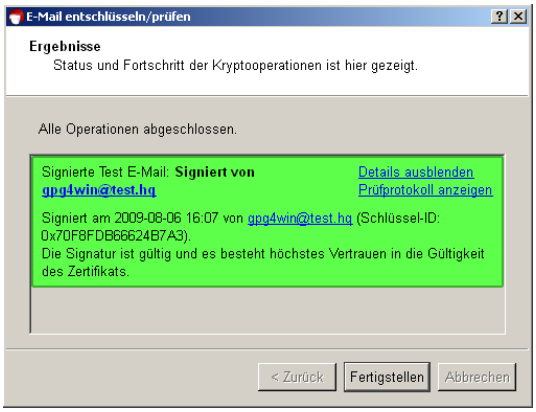
Angenommen Sie erhalten eine signierte E-Mail Ihres Korrespondenzpartners.

Die Überprüfung dieser elektronischen Signatur ist sehr einfach. Alles was Sie dazu brauchen, ist das öffentliche OpenPGP- oder X.509-Zertifikat Ihres Korrespondenzpartners. Dessen öffentliches Zertifikat sollten Sie vor der Überprüfung bereits in Ihre Zertifikatsverwaltung importiert haben (vgl. Kapitel 8).

Um eine signierte OpenPGP- oder S/MIME-E-Mail zu prüfen, gehen Sie genauso vor, wie bei der Entschlüsselung einer E-Mail (vgl. Kapitel 7):

Starten Sie Outlook und öffnen Sie eine signierte E-Mail.

GpgOL übergibt die E-Mail automatisch an Kleopatra zur Prüfung der Signatur. Kleopatra meldet das Ergebnis in einem Statusdialog, z.B.:



Die Signaturprüfung war erfolgreich! Schließen Sie den Dialog, um die signierte E-Mail zu lesen.

Möchten Sie die Überprüfung noch einmal manuell aufrufen, so wählen Sie im Menü der geöffneten E-Mail *Extras→GpgOL Entschlüsseln/Prüfen*.

Sollte die Signaturprüfung fehlt schlagen, wurde die Nachricht bei der Übertragung verändert. Aufgrund der technischen Gegebenheiten im Internet ist es nicht auszuschließen, dass die E-Mail durch eine fehlerhafte Übertragung verändert wurde. Das ist zunächst der wahrscheinlichste Fall. Es kann jedoch auch bedeuten, dass der Text absichtlich verändert wurde.

Wie Sie in einem solchen Fall mit der gebrochenen Signatur umgehen sollten, erfahren Sie im Abschnitt 10.3.

Übrigens...

Wenn Sie kein Gpg4win installiert haben und eine signierte E-Mail öffnen, lässt sich die Signatur natürlich nicht prüfen. Sie sehen dann den E-Mail-Text umrahmt von merkwürdigen Zeilen – das ist die Signatur.

Exemplarisch für OpenPGP sehen Sie, wie dann so eine OpenPGP-signierte E-Mail in Ihrem E-Mail-Programm aussehen würde:

Die E-Mail beginnt mit:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1
```

und endet unter der E-Mail-Nachricht mit:

```
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.2 (MingW32)

iEYEARECAAYFAjxeqy0ACgkQcwePex+3Ivs79wCfW8u
ytRsEXgzCrfPnjGrDDtb7QZIAn17B8l8gFQ3WIUUDCMfA5cQajHcm
=O6lY
-----END PGP SIGNATURE-----
```

Dies ist ein Beispiel – Abwandlungen sind möglich.

10.3 Gründe für eine gebrochene Signatur

Es gibt mehrere Gründe, die zu einem Bruch einer Signatur führen können:

Wenn Sie eine E-Mail mit dem Vermerk „Bad signature“ oder „Überprüfung fehlgeschlagen“ erhalten, ist das ein Warnsignal, dass Ihre E-Mail manipuliert sein könnte! D.h. jemand hat vielleicht den Inhalt oder den Betreff der E-Mail verändert.

Eine gebrochene Signatur muss aber nicht zwangsläufig bedeuten, dass Ihre E-Mail manipuliert wurde. Aufgrund der technischen Gegebenheiten ist es nicht auszuschließen, dass die E-Mail durch eine fehlerhafte Übertragung verändert wurde.

Wichtig: Nehmen Sie eine gebrochene Signatur ernst! Sie sollten in jedem Fall die E-Mail erneut beim Absender anfordern!

Anmerkung:
Es wird grundsätzlich empfohlen Ihr E-Mail-Programm so einzustellen, dass Sie E-Mails nur im „Text“-Format und **nicht** im „HTML“-Format versenden. Sollten Sie dennoch HTML für signierte oder verschlüsselte E-Mails verwenden, können dabei beim Empfänger die Formatierungsinformationen verloren gehen.

Bei Outlook 2003 und 2007 können Sie unter *Extras→Optionen→E-Mail-Format* das Nachrichtenformat auf *Nur Text* umstellen.

10.4 Verschlüsseln und Signieren

Normalerweise verschlüsseln Sie eine Nachricht mit dem öffentlichen Zertifikat Ihres Korrespondenzpartners, der dann mit seinem geheimen Schlüssel die E-Mail entschlüsselt.

Die umgekehrte Möglichkeit – man würde mit dem geheimen Schlüssel verschlüsseln – macht zum Verschlüsseln keinen Sinn, weil alle Welt das dazugehörigen öffentliche Zertifikat kennt und die Nachricht damit entschlüsseln könnte.

Es gibt aber ein anderes Verfahren, um mit Ihrem geheimen Schlüssel eine Datei zu erzeugen: die Signatur (wie Sie am Anfang dieses Kapitels bereits gelesen haben). Solch eine elektronische Signatur bestätigt eindeutig die Urheberschaft – denn wenn jemand Ihr öffentliches Zertifikat auf diese Datei (die Signatur) anwendet und diese Prüfung erfolgreich ist, so kann diese Datei nur von Ihrem privaten Schlüssel kodiert worden sein. Und zu dem dürfen ja nur Sie selbst Zugang haben.

Sie können beide Möglichkeiten kombinieren, also die E-Mail verschlüsseln und signieren:

1. Sie **signieren** die Botschaft mit Ihren eigenen geheimen Schlüssel. Damit ist die Urheberschaft nachweisbar.
2. Dann **verschlüsseln** Sie den Text mit dem öffentlichen Zertifikat des Korrespondenzpartners.

Damit hat die Botschaft sozusagen zwei Sicherheitsmerkmale:

1. Ihren Siegel auf der Nachricht (die Signatur mit Ihrem geheimen Schlüssel).
2. Einen soliden äußeren Umschlag (die Verschlüsselung mit dem öffentlichen Zertifikat des Korrespondenzpartners).

Ihr Korrespondenzpartner öffnet die äußere, starke Hülle mit seinem eigenen geheimen Schlüssel. Hiermit ist die Geheimhaltung gewährleistet, denn nur dieser Schlüssel kann den Text dekodieren. Das Siegel liest er mit Ihrem öffentlichen Zertifikat und hat den Beweis Ihrer Urheberschaft, denn wenn Ihr öffentliches Zertifikat passt, kann er nur mit Ihrem geheimen Schlüssel kodiert worden sein.

Sehr trickreich und – wenn man ein wenig darüber nachdenkt – auch ganz einfach.

11 Wie Sie Ihre E-Mails verschlüsselt archivieren

Eine wichtige Einstellung müssen Sie nun noch vornehmen, um Gpg4win sinnvoll nutzen zu können: Es geht um Ihre E-Mails, die Sie verschlüsselt versenden.

Wie können Sie diese wichtigen Nachrichten sicher archivieren? Natürlich können Sie einfach eine Klartextversion Ihrer Texte aufbewahren, aber das wäre eigentlich nicht angebracht. Wenn Ihre Mitteilung geheimhaltungsbedürftig war, sollte sie auch nicht im Klartext auf Ihrem Rechner gespeichert sein. Sie sollten also stets Ihre verschlüsselt gesendeten E-Mails auch *verschlüsselt* aufbewahren!

Sie ahnen das Problem: Zum Entschlüsseln Ihrer archivierten (versendeten) E-Mails braucht Sie aber den geheimen Schlüssel des Empfängers – und den haben Sie nicht und werden Sie nie haben. . .

Also was tun?

Ganz einfach: **Sie verschlüsseln zusätzlich auch an sich selbst!**

Die Nachricht wird einmal für Ihren eigentlichen Korrespondenzpartner (z.B. Adele) verschlüsselt und ein weiteres Mal auch für Sie, mit Hilfe Ihres eigenen öffentlichen Zertifikats. So können Sie die E-Mail später einfach mit Ihrem eigenen geheimen Schlüssel wieder lesbar machen.

Jede verschlüsselte Nachricht wird von Gpg4win automatisch auch an ihr eigenes Zertifikat verschlüsselt. Dazu nutzt Gpg4win Ihre Absender-E-Mail-Adresse. Sollten Sie mehrere Zertifikate zu einer Adresse besitzen, so müssen Sie sich beim Verschlüsselungsvorgang entscheiden, an welches Zertifikat verschlüsselt werden soll.

Kurz zusammengefasst

1. Sie haben mit dem öffentlichen Zertifikat Ihres Korrespondenzpartners eine E-Mail verschlüsselt und ihm damit geantwortet.
2. Kleopatra verschlüsselt Ihre gesendeten verschlüsselten E-Mails auch zusätzlich mit Ihrem eigenen öffentlichen Zertifikat, so dass die Nachrichten für Sie lesbar bleiben.

Das war's! Zum Ende dieses ersten Teils des Kompendiums werden Sie nun ein sehr fundiertes Einsteigerwissen über Gpg4win besitzen.

Willkommen in der Welt der freien und sicheren E-Mail-Verschlüsselung!

Für ein noch besseres Verständnis, wie Gpg4win im Hintergrund wirklich funktioniert, wird empfohlen, dass Sie sich nun mit dem zweiten, fortgeschrittenen Teil von Gpg4win beschäftigten. Sie werden sehen, dass Sie viele spannende Dinge darin entdecken werden!

Genau wie das Kryptographiesystem Gpg4win wurden dieses Kompendium nicht nur für Mathematiker, Geheimdienstler und Kryptographen geschrieben, sondern **für jedermann.**

Teil II

Für Fortgeschrittene

12 Wie funktioniert Gpg4win?

Das Besondere an Gpg4win und der zugrundeliegenden „Public-Key“ Methode ist, dass sie jeder verstehen kann und soll. Nichts daran ist Geheimwissen – es ist nicht einmal besonders schwer zu verstehen.

Die Benutzung der Gpg4win-Programmkomponenten ist sehr einfach, seine Wirkungsweise dagegen ziemlich kompliziert. Sie werden in diesem Kapitel erklärt bekommen, wie Gpg4win funktioniert – nicht in allen Details, aber so, dass die Prinzipien dahinter deutlicher werden. Wenn Sie diese Prinzipien kennen, werden Sie ein hohes Vertrauen in die Sicherheit von Gpg4win gewinnen.

Am Ende dieses Buches, in Kapitel 24, können Sie – wenn Sie wollen – auch noch die letzten Geheimnisse um die „Public-Key“ Kryptographie lüften und entdecken, warum mit Gpg4win verschlüsselte Nachrichten nach heutigem Stand der Technik nicht zu knacken sind.

Der Herr der Schlüsselringe

Wenn man etwas sehr Wertvolles sichern will, schließt man es am besten ein – mit einem Schlüssel. Noch besser mit einem Schlüssel, den es nur einmal gibt und den man ganz sicher aufbewahrt.

Denn wenn dieser Schlüssel in die falschen Hände fällt, ist es um die Sicherheit des wertvollen Gutes geschehen. Dessen Sicherheit steht und fällt mit der Sicherheit des Schlüssels. Also hat man den Schlüssel mindestens genauso gut abzusichern, wie das zu sichernde Gut selbst. Die genaue Form des Schlüssels muss völlig geheim gehalten werden.

Geheime Schlüssel sind in der Kryptographie ein alter Hut: Schon immer hat man Botschaften geheimzuhalten versucht, indem man den Schlüssel geheimhielt. Dies wirklich sicher zu machen ist sehr umständlich und dazu auch sehr fehleranfällig.

Das Grundproblem bei der „gewöhnlichen“ geheimen Nachrichtenübermittlung ist, dass für Ver- und Entschlüsselung derselbe Schlüssel benutzt wird (symmetrische Verschlüsselung) und dass sowohl der Absender als auch der Empfänger diesen geheimen Schlüssel kennen.

Dies führt zu einer ziemlich paradoxen Situation: Bevor man mit einer solcher Methode ein Geheimnis (eine verschlüsselte Nachricht) mitteilen kann, muss man schon vorher ein anderes Geheimnis (den Schlüssel) mitgeteilt haben. Und da liegt der Hase im Pfeffer: Man muss sich ständig mit dem Problem herumärgern, dass der Schlüssel unbedingt ausgetauscht werden muss, aber auf keinen Fall von einem Dritten abgefangen werden darf.

Gpg4win dagegen arbeitet – außer mit dem geheimen Schlüssel – mit einem weiteren Schlüssel (engl. „key“), der vollkommen frei und öffentlich (engl. „public“) zugänglich ist.

Man spricht daher auch von einem „Public-Key“ Verschlüsselungssystem.

Das klingt widersinnig, ist es aber nicht. Der Witz an der Sache: Es muss kein geheimen Schlüssel mehr ausgetauscht werden. Im Gegenteil: Der geheimen Schlüssel darf auf keinen Fall ausgetauscht werden! Weitergegeben wird nur der öffentliche Schlüssel (im öffentlichen Zertifikat) – und den kann sowieso jeder kennen.

Mit Gpg4win benutzen Sie also ein Schlüsselpaar – einen geheimen und einen zweiten öffentlichen Schlüssel. Beide Schlüssel sind durch eine komplexe mathematische Formel untrennbar miteinander verbunden. Nach heutiger wissenschaftlicher und technischer Kenntnis ist es unmöglich, einen Schlüsselteil aus dem anderen zu berechnen und damit das Verfahren zu knacken. In Kapitel 24 bekommen Sie erklärt, warum das so ist.

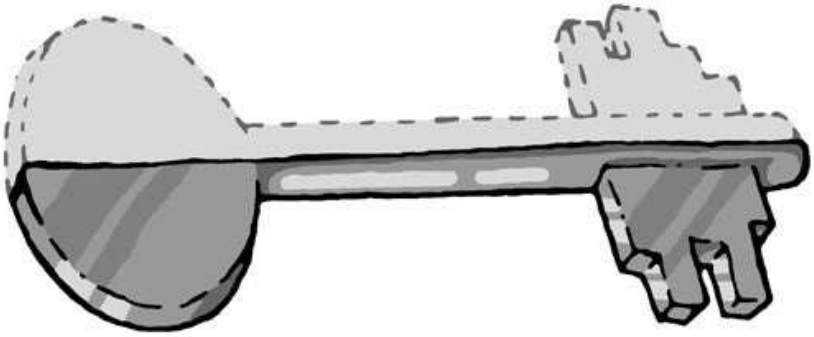
Das Prinzip ist wie gesagt recht einfach:

Der **geheime Schlüssel**, auch **private Schlüssel** genannt (engl. „secret key" oder „private key"), muss geheim gehalten werden.

Der **öffentliche Schlüssel** (engl. „public key"), soll so öffentlich wie möglich gemacht werden.

Beide Schlüsselteile haben ganz und gar unterschiedliche Aufgaben:

Der geheime Schlüsselteil **entschlüsselt** Nachrichten.



Der öffentliche Schlüsselteil **verschlüsselt** Nachrichten.

Der öffentliche Brieftresor

In einem kleinen Gedankenspiel wird die Methode des „Public-Key“ Verschlüsselungssystems und ihr Unterschied zur „Non-Public-Key“ Methode deutlicher...

Die „Non-Public-Key“ Methode geht so:

Stellen Sie sich vor, Sie stellen einen Brieftresor vor Ihrem Haus auf, über den Sie geheime Nachrichten übermitteln wollen.

Der Brieftresor ist mit einem Schloss verschlossen, zu dem es nur einen einzigen Schlüssel gibt. Niemand kann ohne diesen Schlüssel etwas hineinlegen oder herausnehmen. Damit sind Ihre geheimen Nachrichten zunächst einmal gut gesichert.

Da es nur einen Schlüssel gibt, muss Ihr Korrespondenzpartner denselben Schlüssel wie Sie haben, um den Brieftresor damit auf- und zuschließen und eine geheime Nachricht deponieren zu können.

Diesen Schlüssel müssen Sie Ihrem Korrespondenzpartner auf geheimem Wege übergeben.

Erst wenn der andere den geheimen Schlüssel hat, kann er den Brieftresor öffnen und die geheime Nachricht lesen.

Alles dreht sich also um diesen Schlüssel: Wenn ein Dritter ihn kennt, ist es sofort aus mit den geheimen Botschaften. Sie und Ihr Korrespondenzpartner müssen ihn also **genauso** geheim austauschen wie die Botschaft selbst.

Aber – eigentlich könnten Sie ihm bei dieser Gelegenheit ja auch gleich die geheime Mitteilung übergeben…

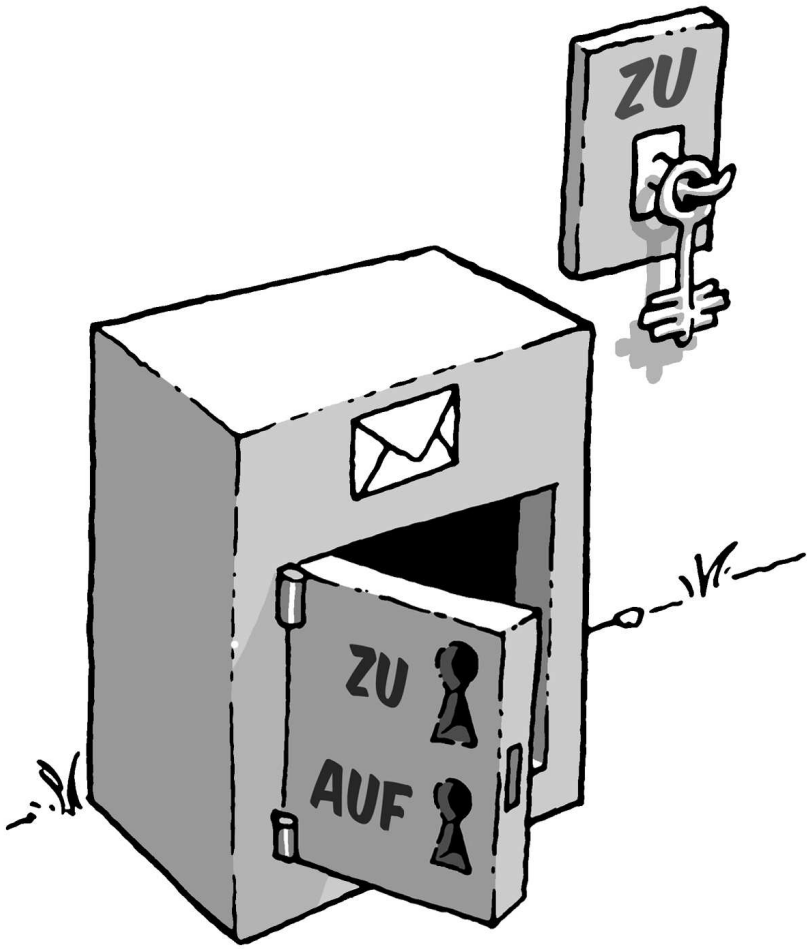
Übertragen auf die E-Mail-Verschlüsselung: Weltweit müssten alle E-Mail-Teilnehmer geheime Schlüssel besitzen und auf geheimem Wege austauschen, bevor sie geheime Nachrichten per E-Mail versenden könnten.

Vergessen Sie diese Möglichkeit am besten sofort wieder…

Nun zur „Public-Key“ Methode:

Sie installieren wieder einen Brieftresor vor Ihrem Haus. Aber: dieser Brieftresor ist – ganz im Gegensatz zu dem ersten Beispiel – stets offen. Direkt daneben hängt – weithin öffentlich sichtbar – ein Schlüssel, mit dem jedermann den Brieftresor zuschließen kann (asymmetrisches Verschlüsselungsverfahren).

Zuschließen, aber nicht aufschließen: das ist der Trick!



Dieser Schlüssel gehört Ihnen, und – Sie ahnen es: Es ist Ihr öffentlicher Schlüssel.

Wenn jemand Ihnen eine geheime Nachricht hinterlassen will, legt er sie in den Brieftresor und schließt mit Ihrem öffentlichen Schlüssel ab. Jedermann kann das tun, denn der Schlüssel dazu ist ja völlig frei zugänglich.

Kein anderer kann den Brieftresor nun öffnen und die Nachricht lesen. Selbst derjenige, der die Nachricht in dem Brieftresor eingeschlossen hat, kann ihn nicht wieder aufschließen, z.B. um die Botschaft nachträglich zu verändern.

Denn die öffentliche Schlüsselhälfte taugt ja nur zum Abschließen.

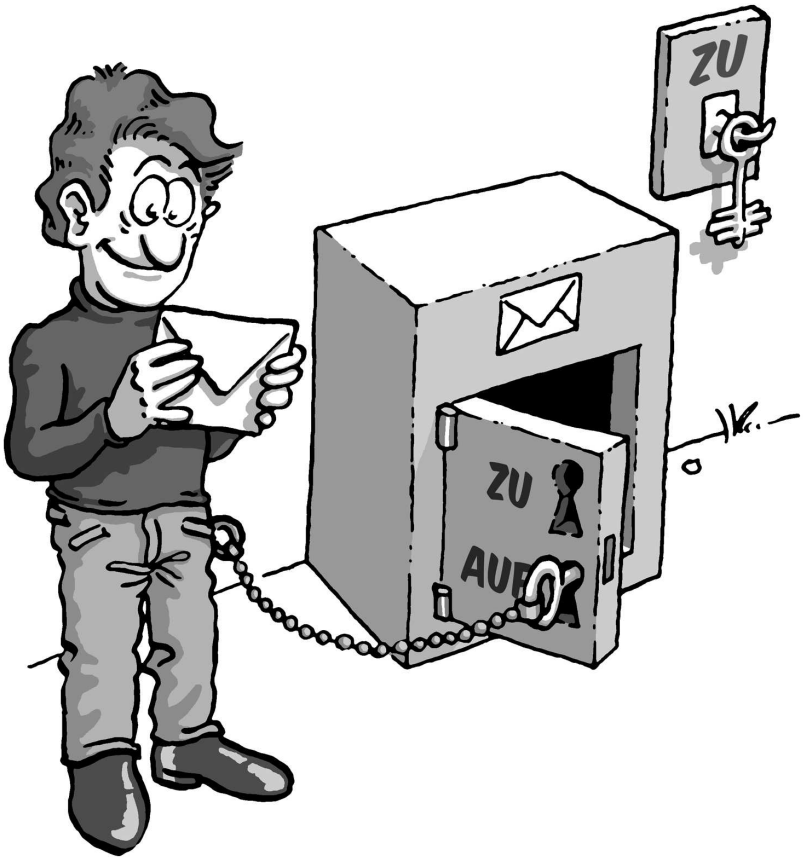
Aufschließen kann man den Brieftresor nur mit einem einzigen Schlüssel: Ihrem eigenen geheimen, privaten Schlüsselteil.

Wieder übertragen auf die E-Mail-Verschlüsselung: Jedermann kann eine E-Mail an Sie verschlüsseln. Er benötigt dazu keineswegs einen geheimen, sondern ganz im Gegenteil einen vollkommen öffentlichen, „ungeheimen“ Schlüssel. Nur ein einziger Schlüssel entschlüsselt die E-Mail wieder: Ihr privater, geheimer Schlüssel.

Spielen Sie das Gedankenspiel noch einmal anders herum:

Wenn Sie einem anderen eine geheime Nachricht zukommen lassen wollen, benutzen Sie dessen Brieftresor mit seinem öffentlichen, frei verfügbaren Schlüssel.

Sie müssen Ihren Briefpartner dazu nicht persönlich kennen, ihn getroffen oder je mit ihm gesprochen haben, denn sein öffentlicher Schlüssel ist überall und jederzeit zugänglich. Wenn Sie Ihre Nachricht hinterlegt und den Brieftresor des Empfängers mit seinem öffentlichem Schlüssel wieder verschlossen haben, ist sie völlig unzugänglich für jeden anderen, auch für Sie selbst. Nur der Empfänger kann den Brieftresor mit seinem privaten Schlüssel öffnen und die Nachricht lesen.



Was ist nun eigentlich gewonnen: Es gibt immer noch einen geheimen Schlüssel!?

Der Unterschied gegenüber der „Non-Public-Key“ Methode ist allerdings ein gewaltiger:

Ihren privater Schlüssel kennen und benutzen nur Sie selbst. Er wird niemals einem Dritten mitgeteilt – die Notwendigkeit einer geheimen Übergabe entfällt, sie verbietet sich sogar.

Es muss überhaupt nichts Geheimes mehr zwischen Absender und Empfänger ausgetauscht werden – weder eine geheime Vereinbarung noch ein geheimes Codewort.

Das ist – im wahrsten Sinne des Wortes – der Knackpunkt: Alle gewöhnlichen (symmetrischen) Verschlüsselungsverfahren können geknackt werden, weil ein Dritter sich beim Schlüsselaustausch in den Besitz des Schlüssels bringen kann.

Dieses Risiko entfällt, weil ein geheimer Schlüssel nicht ausgetauscht wird und sich nur an einem einzigen Ort befindet: dem eigenen Schlüsselbund.

13 Die Passphrase

Wie Sie im letzten Kapitel gelesen haben, ist der private Schlüssel eine der wichtigsten Komponenten in einem „Public-Key“ (asymmetrischen) Verschlüsselungsverfahren. Man muss (und darf) ihn zwar nicht mehr auf geheimem Wege mit seinen Korrespondenzpartnern austauschen, aber nach wie vor ist seine Sicherheit der Schlüssel zur Sicherheit des „ganzen“ Kryptographieverfahrens.

Es ist deswegen eminent wichtig, diesen privaten Schlüssel sicher abzuspeichern. Dies geschieht auf zweierlei Weise:

Jeder andere Benutzer des Rechners, auf dessen Festplatte dieser Schlüssel gespeichert ist, darf keinen Zugriff auf ihn erhalten – weder zum Schreiben noch zum Lesen. Es ist deswegen unbedingt zu vermeiden, den Schlüssel in einem öffentlichen Dateiordner (z.B. `C:\Temp` oder `C:\WINNT`) abzulegen. Gpg4win speichert den Schlüssel deswegen im sogenannten „Heimatverzeichnis“ („Eigene Dateien“) von GnuPG ab. Dies kann sich je nach konfiguriertem Betriebssystem an unterschiedlichen Orten befinden; für einen Benutzer mit dem Anmeldenamen „Harry“ könnte es z.B.:
`C:\Dokumente und Einstellungen\harry\Anwendungsdaten\gnupg`
sein. Der geheime Schlüssel befindet sich dort in einer Datei mit dem Namen `secring.gpg`.

Dieser Schutz allein ist allerdings nicht ausreichend: Zum einen kann der Administrator des Rechners immer auf alle Dateien zugreifen – also auch auf Ihren geheimen Schlüssel. Zum anderen könnte der Rechner abhanden kommen oder durch „Malware“ (Viren-, Würmer-, Trojanersoftware) kompromittiert werden.

Ein weiterer Schutz ist deswegen notwendig. Dieser besteht aus einer Passphrase.

Die Passphrase sollte nicht nur aus einem Wort bestehen, sondern z.B. aus einem Satz. Sie sollten diese Passphrase wirklich „im Kopf“ behalten und niemals aufschreiben müssen.

Trotzdem darf sie nicht erraten werden können. Das klingt vielleicht widersprüchlich, ist es aber nicht. Es gibt einige erprobte Tricks, mit deren Hilfe Sie sich eine völlig individuelle, leicht zu merkende und nur sehr schwer zu erratende Passphrase ausdenken können.

Eine **gute Passphrase** kann so entstehen:

Denken Sie an einen Ihnen gut bekannten Satz, z.B.:

```
Ein blindes Huhn findet auch einmal ein Korn.
```

Aus diesem Satz nehmen Sie beispielsweise jeden dritten Buchstaben:

```
nieufdahnlnr (Ein blindes Huhn findet auch einmal ein Korn.)
```

Diesen Buchstabensalat können Sie sich zunächst sicher nicht gut merken, aber Sie werden ihn eigentlich nie vergessen, solange Sie den ursprünglichen Satz im Kopf haben. Im Laufe der Zeit und je öfter Sie ihn benutzen, prägt sich so eine Passphrase in Ihr Gedächtnis. Erraten kann diese Passphrase niemand.

Denken Sie an ein Ereignis, das sich bereits fest in Ihrem persönlichen Langzeitgedächtnis verankert hat. Vielleicht gibt es einen Satz, mit dem sich Ihr Kind oder Ihr Partner „unvergesslich“ gemacht hat. Oder eine Ferienerinnerung, oder einer Textzeile aus einem für Sie wichtigen Liedes.

Verwenden Sie kleine und große Buchstaben, Nummern, Sonder- und Leerzeichen durcheinander. Im Prinzip ist alles erlaubt, auch „Ö“, „ß“, „$“ usw.

Aber Vorsicht – falls Sie Ihren geheimen Schlüssel im Ausland an einem fremden Rechner benutzen wollen, bedenken Sie, dass fremdsprachige Tastaturen diese Sonderzeichen oft nicht haben. Beispielsweise werden Sie kein „ä“ auf einer englischen Tastatur finden.

Machen Sie Rechtschreibfehler, z.B. „feLer“ statt „Fehler“. Natürlich müssen Sie sich diese „feLer“ gut merken können. Oder wechseln Sie mittendrin die Sprache. Aus dem schönen Satz:

```
In München steht ein Hofbräuhaus.
```

könnten man beispielsweise diese Passphrase machen:

```
inMinschen stet 1h0f breuhome
```

Denken Sie sich einen Satz aus, der möglichst unsinnig ist, den Sie sich aber doch merken können, wie z.B.:

```
Es blaut so garstig beim Walfang, neben Taschengeld, auch im
Winter.
```

Eine Passphrase in dieser Länge ist ein sicherer Schutz für Ihren geheimen Schlüssel.

Sie darf auch kürzer sein, wenn Sie einige Buchstaben groß schreiben, z.B. so:

```
Es blAut nEBen TaschengeLd auch im WiNter.
```

Das ist nun kürzer, aber nicht mehr so leicht zu merken. Wenn Sie eine noch kürzere Passphrase verwenden, indem Sie hier und da Sonderzeichen benutzen, haben Sie zwar bei der Eingabe weniger zu tippen, aber die Wahrscheinlichkeit, dass Sie Ihre Passphrase vergessen, wird dabei noch größer.

Ein extremes Beispiel für eine möglichst kurze, aber dennoch sehr sichere Passphrase ist dieses hier:

```
R!Qw"s,UIb *7\$
```

In der Praxis haben sich solche Zeichenfolgen allerdings als recht wenig brauchbar herausgestellt, da man einfach zu wenig Anhaltspunkte für die Erinnerung hat.

Eine **schlechte Passphrase** ist blitzschnell „geknackt“, wenn sie...

- ... schon für einen anderen Zweck benutzt wird (z.B. für einen E-Mail-Account oder Ihr Handy). Die gleiche Passphrase wäre damit bereits einer anderen, möglicherweise unsicheren Software bekannt.
- ... aus einem Wörterbuch stammt. Passphrase-Knackprogramme prüfen binnen Minuten umfangreiche elektronische Wörterbücher.
- ... aus einem Geburtsdatum, einem Namen oder anderen öffentlichen Informationen besteht. Wer vorhat, Ihre E-Mail zu entschlüsseln, wird sich diese Daten beschaffen.
- ... ein landläufiges Zitat ist; wie z.B. „das wird böse enden“ oder „to be or not to be“. Auch mit derartigen gängigen Zitaten testen Passphrase-Knackprogramme routinemäßig und blitzschnell eine Passphrase.
- ... aus nur einem Wort oder aus weniger als 8 Zeichen besteht. Denken Sie sich unbedingt eine längere Passphrase aus.

Wenn Sie nun Ihre Passphrase zusammenstellen, nehmen Sie **auf gar keinen Fall** eines der oben angeführten Beispiele. Denn es liegt auf der Hand: Wenn sich jemand ernsthaft darum bemüht Ihre Passphrase herauszubekommen, würde er zuerst ausprobieren, ob Sie nicht eines dieser Beispiele genommen haben.

Seien Sie kreativ! Denken Sie sich jetzt eine Passphrase aus! Unvergesslich und unknackbar.

Lesen Sie dann im Kapitel 5 weiter, um Ihre neue Passphrase bei der Erzeugung Ihres Schlüsselpaars festzulegen.

14 Zertifikat im Detail

Auf Seite 43 haben Sie sich schon den Detaildialog Ihres erzeugten Zertifikats angesehen. Viele Angaben zu Ihrem Zertifikat sind dort aufgelistet. Im folgenden Abschnitt bekommen Sie einen Überblick über die wichtigsten Punkte (beachten Sie dabei die Unterschiede für OpenPGP- und X.509-Zertifikate):

- die Benutzer-Kennung
- den Fingerabdruck
- die Gültigkeit
- das Vertrauen in den Zertifikatsinhaber **(nur OpenPGP)**
- die Beglaubigungen **(nur OpenPGP)**

Die Benutzer-Kennung besteht aus dem Namen und der E-Mail-Adresse, die Sie während der Zertifikatserzeugung eingegeben haben, also z.B.:
`Heinrich Heine <heinrichh@gpg4win.de>`

Für OpenPGP-Zertifikate können Sie mit Kleopatra über den Menüpunkt *Zertifikate → Benutzer-Kennung hinzufügen...* Ihr Zertifikat um weitere Benutzer-Kennungen erweitern. Das ist dann sinnvoll, wenn Sie z.B. für eine weitere E-Mail-Adressen das selbe Zertifikat nutzen möchten.

Beachten Sie: Hinzufügen neuer Benutzer-Kennungen ist in Kleopatra nur für OpenPGP-Zertifikate (nicht aber für X.509) möglich.

Der Fingerabdruck wird verwendet, um mehrere Zertifikate voneinander zu unterscheiden. Mit dieser Kennung können Sie nach (öffentlichen) Zertifikaten suchen, die z.B. auf einem weltweit verfügbaren OpenPGP-Zertifikatsservern (engl. „key server“) oder auf einem X.509-Zertifikatsserver liegen. Was Zertifikatsserver sind, erfahren Sie im folgenden Kapitel.

Die Gültigkeit von Zertifikaten wird stets unter dem zeitlichen Aspekt betrachtet. Für OpenPGP-Zertifikate ist die Gültigkeit normalerweise auf „Unbegrenzt“ gesetzt. Sie können dies selbständig mit Kleopatra ändern, indem Sie auf die Schaltfläche „Ablaufdatum ändern“ in den Zertifikatsdetails klicken (oder den Menü *Zertifikate→Ablaufdatum ändern* auswählen) und ein neues Datum eintragen. Damit können Sie Zertifikate als nur für eine begrenzte Zeit gültig erklären, z.B. um sie an externe Mitarbeiter auszugeben.

Die Gültigkeit von X.509-Zertifikaten wird bei der Zertifikatsausstellung von der Beglaubigungsinstanz (CA) festgelegt und kann nicht vom Nutzer geändert werden.

Das Vertrauen in den Zertifikatsinhaber beschreibt das Maß an Zuversicht, das Sie subjektiv in den Besitzer des OpenPGP-Zertifikats setzen, andere OpenPGP-Zertifikate korrekt zu beglaubigen. Sie können das Vertrauen über die Schaltfläche [*Vertrauen in den Zertifikatsinhaber ändern*] in den Zertifikatsdetails (oder über das Menü *Zertifikate→Vertrauensstatus ändern*) einstellen.

Beachten Sie: Der Vertrauensstatus ist nur für OpenPGP-Zertifikate relevant. Für X.509-Zertifikate gibt es diese Vertrauensmethode nicht.

Die Beglaubigungen Ihres OpenPGP-Zertifikats beinhalten die Benutzer-Kennungen derjenigen Zertifikatsinhaber, die sich von der Echtheit Ihres Zertifikats überzeugt und es dann auch beglaubigt haben. Das Vertrauen in die Echtheit Ihres Zertifikats steigt mit der Anzahl an Beglaubigungen, die Sie von anderen Nutzern erhalten.

Beachten Sie: Beglaubigungen sind nur für OpenPGP-Zertifikate relevant. Für X.509-Zertifikate gibt es diese Vetrauensmethode nicht.

Diese Zertifikatsdetails sind für die tagtägliche Benutzung von Gpg4win nicht unbedingt erforderlich. Sie werden relevant, wenn Sie neue Zertifikate erhalten oder ändern.

Wie Sie fremde Zertifikate prüfen und beglaubigen und was genau das „Netz des Vertrauens“ ist, erfahren Sie im Kapitel 16.

15 Die Zertifikatsserver

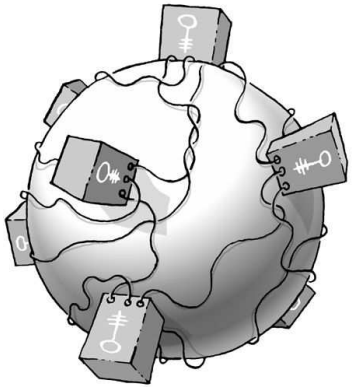
Die Nutzung eines Zertifikatsservers zum Verbreiten Ihres öffentlichen (OpenPGP- oder X.509-) Zertifikats wurde bereits im Abschnitt 6.2 einführend erläutert. Dieses Kapitel beschäftigt sich mit den Details von Zertifikatsservern und zeigt Ihnen, wie sie diese mit Kleopatra nutzen können.

Zertifikatsserver können von allen Programmen benutzt werden, die den OpenPGP- bzw. X.509-Standard unterstützen.

Kleopatra unterstützt zwei Arten von Zertifikatsservern:

- OpenPGP-Zertifikatsserver
- X.509-Zertifikatsserver

OpenPGP-Zertifikatsserver (im Englischen auch „key server“ genannt) sind dezentral organisiert und synchronisieren sich weltweit miteinander. Aktuelle Statistiken über ihre Zahl oder die Anzahl der dort liegenden OpenPGP-Zertifikate gibt es nicht. Dieses verteilte Netz von OpenPGP-Zertifikatsservern sorgt für eine bessere Verfügbarkeit und verhindert, dass einzelne Systemadministratoren Zertifikate löschen, um so die Kommunikation unmöglich zu machen („Denial of Service“-Angriff).



X.509-Zertifikatsserver werden in der Regel von den Beglaubigungsinstanzen (CAs) über LDAP bereitgestellt und werden manchmal auch als Verzeichnisdienste für X.509-Zertifikate bezeichnet.

15.1 Zertifikatsserver einrichten

Öffnen Sie den Konfigurationsdialog von Kleopatra: *Einstellungen→Kleopatra einrichten...*

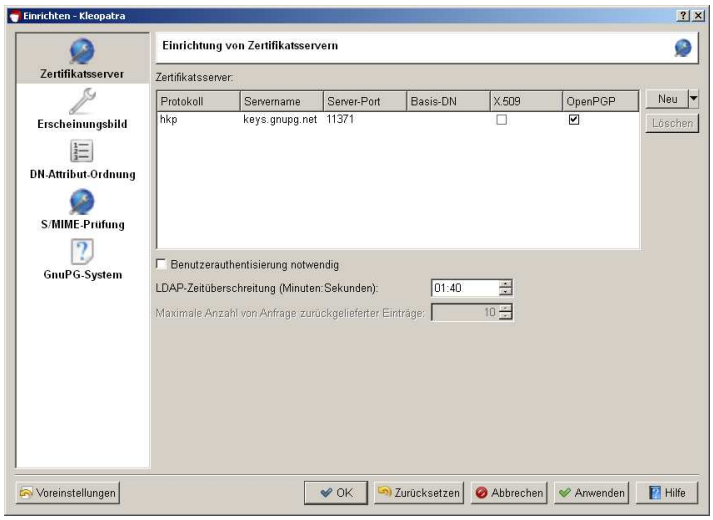
Legen Sie unter der Gruppe *Zertifikatsserver* einen neuen Zertifikatsserver an, indem Sie auf die Schaltfläche *Neu* klicken. Wählen Sie zwischen *OpenPGP* oder *X.509*.

Bei *OpenPGP* wird in die Liste ein voreingestellter OpenPGP-Zertifikatsserver mit der Serveradresse `hkp://keys.gnupg.net` (Port: 11371, Protokoll: hkp) hinzugefügt. Sie können diesen ohne Änderung direkt verwenden - oder Sie nutzen eine der vorgeschlagenen OpenPGP-Serveradressen von der nächsten Seite.

Bei *X.509* erhalten Sie folgende Vorbelegungen für einen X.509-Zertifikatsserver: (Protokoll: ldap, Severname: server, Server-Port: 389). Vervollständigen Sie die Angaben zu Servername und Basis-DN Ihres X.509-Zertifikatsservers und prüfen Sie den Server-Port.

Sollte Ihr Zertifikatsserver Benutzername und Passwort fordern, so aktivieren Sie die Option *Benutzerauthentisierung notwendig* und tragen Ihre gewünschten Angaben ein.

Der folgende Screenshot zeigt einen konfigurierten OpenPGP-Zertifikatsserver:



Bestätigen Sie abschließend Ihre Konfiguration mit [*OK*]. Ihr Zertifikatsserver ist nun erfolgreich eingerichtet.

Um sicherzugehen, dass Sie den Zertifikatsserver korrekt konfiguriert haben, ist es hilfreich z.B. eine Zertifikatssuche auf dem Server zu starten (Anleitung siehe Abschnitt).

Proxy-Einstellung: Falls Sie einen Proxy in Ihrem Netzwerk nutzen, sollten Sie die Zertifikatsserver-Adresse in der Spalte „Servername“ um den Parameter `http-proxy=<proxydomain>` ergänzen. Der vollständige Servername könnte also z.B. lauten:
`keys.gnupg.net http-proxy=proxy.hq`
Kontrollieren und ggf. korrigieren können Sie die Zertifikatsserver-Konfigurationen auch in der Datei:
`%APPDATA%\gnupg\gpg.conf`

OpenPGP-Zertifikatsserver-Adressen

Es wird empfohlen, nur moderne OpenPGP-Zertifikatsserver zu verwenden, da nur diese mit den neueren Merkmalen von OpenPGP umgehen können.

Hier eine Auswahl von gut funktionierenden Zertifikatsservern:

- hkp://blackhole.pca.dfn.de
- hkp://pks.gpg.cz
- hkp://pgp.cns.ualberta.ca
- hkp://minsky.surfnet.nl
- hkp://keyserver.ubuntu.com
- hkp://keyserver.pramberger.at
- http://keyserver.pramberger.at
- http://gpg-keyserver.de

Sollte Sie Probleme mit einer Firewall haben, so versuchen Sie am besten die Zertifikatsserver, deren URL mit `http://` beginnen.

Die Zertifikatsserver unter den Adressen

- hkp://keys.gnupg.net (Vorauswahl von Kleopatra, siehe Bildschirmfoto auf vorheriger Seite)
- hkp://subkeys.pgp.net

sind ein Sammelpunkt für ein ganzes Netz dieser Server; es wird dann zufällig ein konkreter Server ausgewählt.

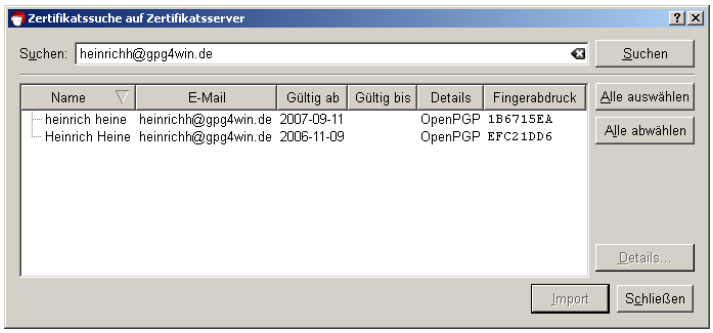
Achtung: Nicht `ldap://keyserver.pgp.com` als Zertifikatsserver benutzen, weil er sich nicht mit den anderen Servern synchronisiert (Stand: August 2009).

15.2 Zertifikate auf Zertifikatsserver suchen und importieren

Nachdem Sie mindestens einen Zertifikatsserver eingerichtet haben, können Sie nun dort nach Zertifikaten suchen und diese anschließend importieren.

Klicken Sie dazu in Kleopatra auf *Datei→Zertifikate auf Server suchen...*.

Sie erhalten einen Suchdialog, in dessen Eingabefeld Sie den Namen des Zertifikatsbesitzers, oder eindeutiger und daher besser geeignet, seine E-Mail-Adresse oder den Fingerabdruck seines Zertifikats eingeben können.



Um die Details eines ausgewählten Zertifikats zu sehen, klicken Sie auf die Schaltfläche [*Details...*].

Möchten Sie nun eines der gefundenen Zertifikate in Ihre lokale Zertifikatssammlung hinzufügen? Dann selektieren Sie das Zertifikat aus der Liste der Suchergebnisse und klicken Sie auf [*Importieren*].

Kleopatra zeigt Ihnen anschließend einen Dialog mit den Ergebnissen des Importvorgangs an. Bestätigen Sie diesen mit [*OK*].

War der Import erfolgreich, finden Sie nun das ausgewählte Zertifikat in der Zertifikatsverwaltung von Kleopatra.

15.3 Zertifikate auf OpenPGP-Zertifikatsserver exportieren

Wenn Sie einen OpenPGP-Zertifikatsserver eingerichtet haben (siehe Abschnitt 15.1), genügt ein Mausklick und Ihr öffentliches OpenPGP-Zertifikat ist unterwegs rund um die Welt:

Wählen Sie Ihr OpenPGP-Zertifikat in Kleopatra aus und klicken anschließend auf den Menüeintrag: *Datei→Zertifikate nach Server exportieren...*.

Sie brauchen Ihr Zertifikat nur an irgendeinen der verfügbaren OpenPGP-Zertifikatsserver zu senden, denn fast alle synchronisieren sich weltweit miteinander. Es kann ein, zwei Tage dauern, bis Ihr OpenPGP-Zertifikat wirklich überall verfügbar ist, aber dann haben Sie ein „globales“ Zertifikat.

Sollten Sie Ihr Zertifikat exportieren ohne zuvor einen OpenPGP-Zertifikatsserver eingerichtet zu haben, so schlägt Ihnen Kleopatra den bereits voreingestellten Server `hkp://keys.gnupg.net` zur Verwendung vor.

16 Die Zertifikatsprüfung

Woher wissen Sie eigentlich, dass das fremde Zertifikat wirklich vom genannten Absender stammt? Und umgekehrt – warum sollte Ihr Korrespondenzpartner glauben, dass das Zertifikat, das Sie ihm geschickt haben, auch wirklich von Ihnen stammt? Die Absenderangabe auf einer E-Mail besagt eigentlich gar nichts, genauso wie die Absenderangabe auf einem Briefumschlag.

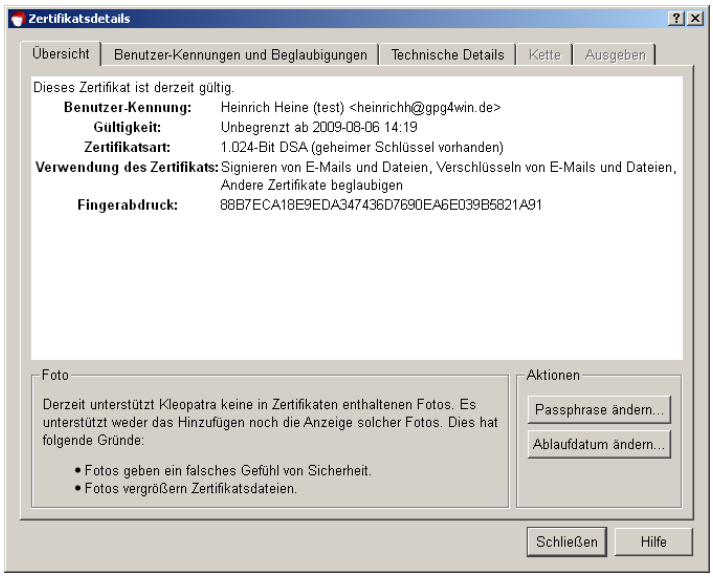
Wenn Ihre Bank z.B. eine E-Mail mit Ihrem Namen und der Anweisung erhält, Ihr sämtliches Guthaben auf ein Nummernkonto auf den Bahamas zu überweisen, wird sie sich hoffentlich weigern – E-Mail-Adresse hin oder her. Eine E-Mail-Adresse besagt überhaupt nichts über die Identität des Absenders.

Der Fingerabdruck

Wenn Sie nur einen kleinen Kreis von Korrespondenzpartnern haben, ist die Sache mit der Identität schnell geregelt: Sie prüfen den Fingerabdruck des anderen Zertifikats.

Jedes Zertifikat trägt eine einmalige Kennzeichnung, die es zweifelsfrei identifiziert; besser noch als ein Fingerabdruck eines Menschen. Deshalb bezeichnet man diese Kennzeichnung eben als Fingerabdruck.

Wenn Sie sich zu einem Zertifikat die Details in Kleopatra anzeigen lassen (z.B. durch Doppelklick auf das Zertifikat), sehen Sie u.a. dessen 40-stelligen Fingerabdruck:



Der Fingerabdruck des oben dargestellten OpenPGP-Zertifikats ist also:
`88B7ECA18E9EDA347436D7690EA6E039B5821A91`

Wie gesagt – der Fingerabdruck identifiziert das Zertifikat und seinen Besitzer eindeutig.

Rufen Sie Ihren Korrespondenzpartner einfach an, und lassen Sie sich von ihm den Fingerabdruck seines Zertifikats vorlesen. Wenn die Angaben mit dem Ihnen vorliegenden Zertifikat übereinstimmen, haben Sie eindeutig das richtige Zertifikat.

Natürlich können Sie sich auch persönlich mit dem Eigentümer des Zertifikats treffen oder auf jedem anderen Wege sicher stellen, dass Zertifikat und Eigentümer zusammen gehören. Häufig ist der Fingerabdruck auch auf Visitenkarten abgedruckt; wenn Sie also eine authentische Visitenkarte haben, so können Sie sich den Anruf ersparen.

OpenPGP-Zertifikat beglaubigen

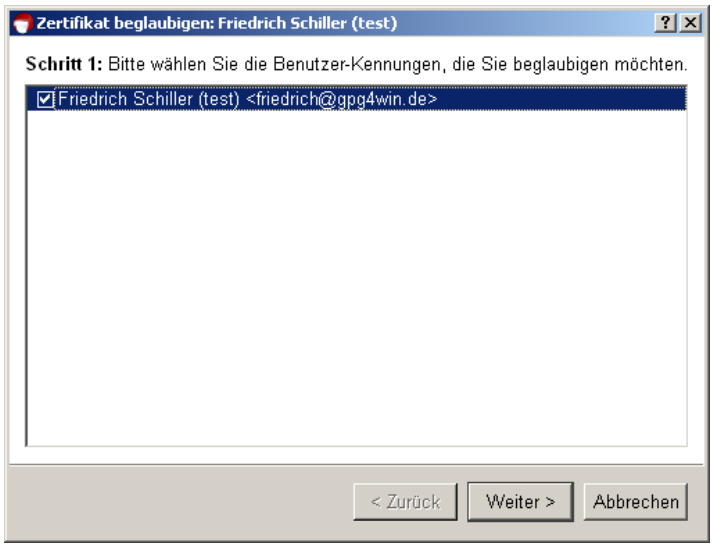
Nachdem Sie sich „per Fingerabdruck“ von der Echtheit des Zertifikats überzeugt haben, haben Sie nun die Möglichkeit, dieses Zertifikat zu beglaubigen.

Beachten Sie: Beglaubigen von Zertifikaten durch Benutzer ist nur mit OpenPGP möglich. Bei X.509 ist das Beglaubigungsinstanzen (CAs) vorbehalten!

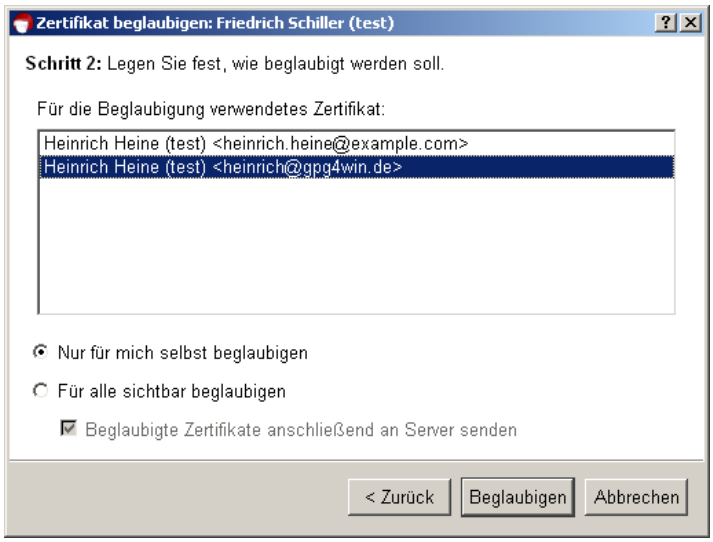
Durch das Beglaubigen eines (fremden) Zertifikats teilen Sie anderen (Gpg4win-)Benutzern mit, dass Sie dieses Zertifikat für echt halten: Sie übernehmen so etwas wie die „Patenschaft“ über dieses Zertifikat und erhöhen das allgemeine Vertrauen in seiner Echtheit.

Wie funktioniert das Beglaubigen nun genau?
Selektieren Sie in Kleopatra das OpenPGP-Zertifikat, dass Sie für echt halten und beglaubigen möchten. Wählen Sie anschließend im Menü: *Zertifikate→Zertifikat beglaubigen...*

Im nachfolgenden Dialog bestätigen Sie nun noch einmal das zu beglaubigende OpenPGP-Zertifikat mit [*Weiter*]:



Im nächsten Schritt wählen Sie Ihr eigenes OpenPGP-Zertifikat aus, mit dem Sie das (im letzten Schritt ausgewählte) Zertifikat beglaubigen wollen:

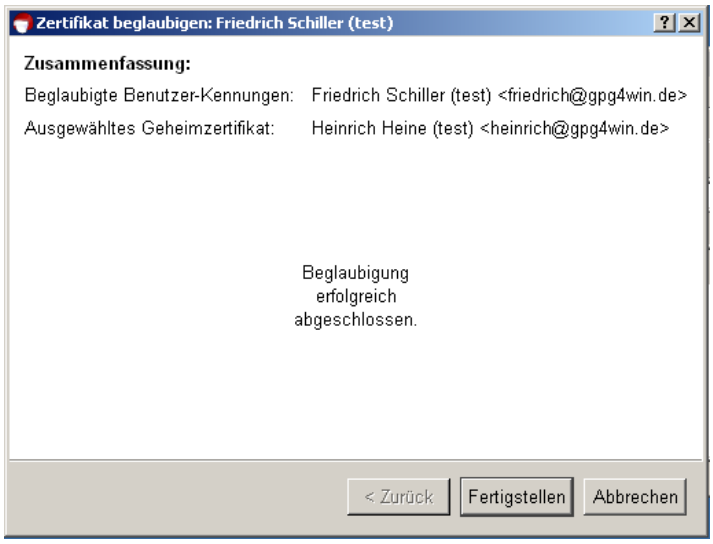


Entscheiden Sie hier, ob Sie [*Nur für mich selbst beglaubigen*] oder [*Für alle sichtbar beglaubigen*] wollen. Bei letzterer Variante haben Sie die Option, das beglaubigte Zertifikat anschließend auf einen OpenPGP-Zertifikatsserver hochzuladen und damit der Welt ein mit Ihrer Beglaubigung versehenes, aktualisiertes Zertifikat zur Verfügung zu stellen.

Bestätigen Sie Ihre Auswahl mit [*Beglaubigen*].

Wie beim Signieren einer E-Mail müssen Sie auch beim Beglaubigen eines Zertifikats (mit Ihrem privaten Schlüssel) Ihre Passphrase eingeben. Erst nach korrekter Eingabe ist die Beglaubigung abgeschlossen.

Nach erfolgreicher Beglaubigung erhalten Sie folgendes Fenster:



Wollen Sie die erfolgte Beglaubigung nun einmal prüfen?
Dann öffnen Sie die Zertifikatsdetails des eben beglaubigten Zertifikats. Wählen Sie den Reiter *Benutzer-Kennungen und Beglaubigungen* und klicken auf die Schaltfläche [*Hole Beglaubigungen ein*].

Sortiert nach den Benutzer-Kennungen sehen sie alle Beglaubigungen, die in diesem Zertifikat enthalten sind. Hier sollte Sie auch Ihr Zertifikat wiederfinden, mit dem Sie eben beglaubigt haben.

Das Netz des Vertrauens

Durch das Beglaubigen von Zertifikaten entsteht – auch über den Kreis von Gpg4win-Benutzern und Ihrer täglichen Korrespondenz hinaus – ein „Netz des Vertrauens“, bei dem Sie nicht mehr zwangsläufig darauf angewiesen sind, ein OpenPGP-Zertifikat direkt zu prüfen.

Natürlich steigt das Vertrauen in ein Zertifikat, wenn mehrere Leute es beglaubigen. Ihr eigenes OpenPGP-Zertifikat wird im Laufe der Zeit die Beglaubigungen vieler anderer GnuPG-Benutzer tragen. Damit können immer mehr Menschen darauf vertrauen, dass dieses Zertifikat wirklich Ihnen und niemandem sonst gehört.

Wenn man dieses „Web of Trust“ weiterspinnt, entsteht eine flexible Beglaubigungs-Infrastruktur.

Eine einzige Möglichkeit ist denkbar, mit dem man diese Zertifikatsprüfung aushebeln kann: Jemand schiebt Ihnen einen falsches Zertifikat unter. Also einen öffentlichen OpenPGP-Schlüssel, der vorgibt, von X zu stammen, in Wirklichkeit aber von Y ausgetauscht wurde. Wenn ein solches gefälschtes Zertifikat beglaubigt wird, hat das „Netz des Vertrauens“ natürlich ein Loch. Deshalb ist es so wichtig, sich zu vergewissern, ob ein Zertifikat, wirklich zu der Person gehört, der es zu gehören vorgibt.

Was aber, wenn eine Bank oder Behörde prüfen möchte, ob die Zertifikate ihrer Kunden echt sind? Alle anzurufen, kann hier sicher nicht die Lösung sein…

Beglaubigungsinstanzen

Hier braucht man eine „übergeordnete“ Instanz, der alle Benutzer vertrauen können. Sie prüfen ja auch nicht persönlich den Personalausweis eines Unbekannten durch einen Anruf beim Einwohnermeldeamt, sondern vertrauen darauf, dass die ausstellende Behörde diese Überprüfung korrekt durchgeführt und beglaubigt hat.

Solche Beglaubigungsinstanzen gibt es auch für OpenPGP. In Deutschland bietet unter anderem z.B. die Zeitschrift c’t schon lange einen solchen Dienst kostenlos an, ebenso wie viele Universitäten.

Wenn man also ein OpenPGP-Zertifikat erhält, das durch den Zertifikatsaussteller per Beglaubigung seine Echtheit bestätigt, kann man sich darauf verlassen.

Derartige Beglaubigungsinstanzen oder „Trust Center“ sind auch bei anderen Verschlüsselungsverfahren – wie z.B. S/MIME – vorgesehen. Im Gegensatz zum „Web of Trust” sind sie hierarchisch strukturiert: Es gibt eine „Oberste Beglaubigungsinstanz“, die weitere „Unterinstanzen“ beglaubigt und ihnen das Recht vergibt, Benutzerzertifikate zu beglaubigen (vgl. Kapitel 3).

Am besten ist diese Infrastruktur mit einem Siegel vergleichbar: Die Plakette auf Ihrem Autonummernschild kann Ihnen nur eine dazu berichtigte Institution geben, die die Befugnis dazu wiederum von einer übergeordneten Stelle erhalten hat. Technisch ist eine Beglaubigung nichts anderes als eine Signatur eines Zertifikates durch den Beglaubigenden.

Mit der hierarchischen Beglaubigungs-Infrastruktur entspricht dieses Modell natürlich wesentlich besser den Bedürfnissen staatlicher und behördlicher Instanzen als das lose, auf gegenseitigem Vertrauen beruhende „Web of Trust“ von GnuPG. Der Kern der Beglaubigung selbst ist allerdings völlig identisch: Gpg4win unterstützt neben dem „Web of Trust“ (OpenPGP) zusätzlich auch eine hierarchische Beglaubigungsstruktur (S/MIME). Demnach bietet Gpg4win eine Grundlage um dem strengen Signaturgesetz der Bundesrepublik Deutschland zu entsprechen.

Wenn Sie sich weiter für dieses Thema interessieren, dann können Sie sich z.B. bei folgenden Webadressen über dieses und viele andere IT-Sicherheits-Themen informieren:

- `www.bsi.de` des BSIs,
- `www.bsi-fuer-buerger.de`
- `www.gpg4win.de`

Eine weitere exzellente, mehr technische Informationsquelle zum Thema der Beglaubigungsinfrastrukturen bietet das `Original GnuPG Handbuch`, das Sie ebenfalls im Internet finden (`www.gnupg.org/gph/de/manual`).

17 Dateianhänge verschlüsseln

Wenn Sie eine verschlüsselte E-Mail versenden und Dateien anhängen, so wollen Sie in der Regel sicherlich auch, dass diese Anhänge verschlüsselt werden.

Bei einer komfortablen Integration von GnuPG in Ihr E-Mail-Programm sollten Anhänge genauso behandelt werden wie der eigentlichen Text Ihrer E-Mail, also signiert, verschlüsselt oder beides zusammen.

GpgOL übernimmt die Verschlüsselung und Signierung von Anhängen automatisch.

Bei weniger komfortabel in einem E-Mail-Programm integrierten Verschlüsselungswerkzeug müssen Sie aufpassen: Die Anhänge werden oft unverschlüsselt mitgesendet.

Was kann man in so einem Fall tun? Ganz einfach: Sie verschlüsseln den Anhang getrennt und hängen ihn dann in verschlüsseltem Zustand an die E-Mail an. Es läuft also auf ein ganz gewöhnliches Verschlüsseln von Dateien hinaus, das in Kapitel 18 beschrieben ist.

18 Dateien signieren und verschlüsseln

Nicht nur E-Mails, sondern auch einzelne Dateien können Sie mit Gpg4win signieren und verschlüsseln. Das Prinzip ist das gleiche:

- Sie **signieren** eine Datei mit Hilfe Ihres Zertifikats, um sicherzugehen, dass die Datei unverändert bei Ihrem Empfänger ankommt.
- Sie **verschlüsseln** eine Datei mit Hilfe des Zertifikat des Empfängers, um die Datei vor unbefugten Personen geheim zu halten.

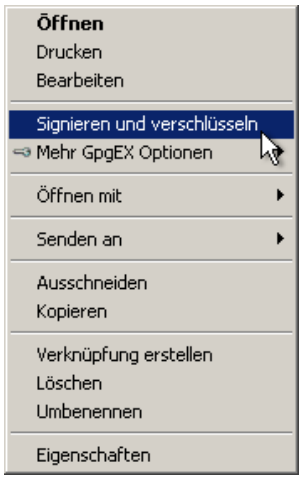
Mit der Anwendung **GpgEX** können Sie Dateien ganz einfach aus Ihrem Windows Explorer heraus signieren oder verschlüsseln – egal ob OpenPGP oder S/MIME. Dieses Kapitel erläutert Ihnen, wie das genau funktioniert.

Sollten Sie eine Datei als E-Mail-Anhang verschicken, übernimmt z.B. GpgOL automatisch die Signierung bzw. Verschlüsselung der Datei zusammen mit Ihrer E-Mail. Sie brauchen sich in diesem Fall nicht gesondert darum kümmern.

18.1 Dateien signieren und prüfen

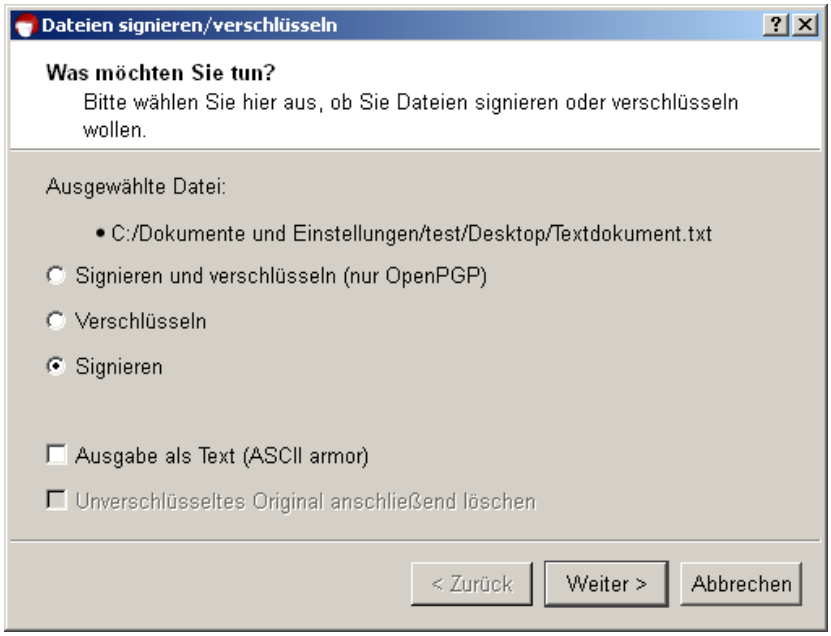
Beim Signieren einer Datei kommt es vorrangig nicht auf die Geheimhaltung, sondern auf die Unveränderheit (Integrität) der Datei an.

Die Signierung können Sie bequem mit **GpgEX** aus dem Kontextmenü des Windows Explorer ausführen. Selektieren Sie eine (oder mehrere) Datei(en) und öffnen Sie mit der rechten Maustaste das Kontextmenü:



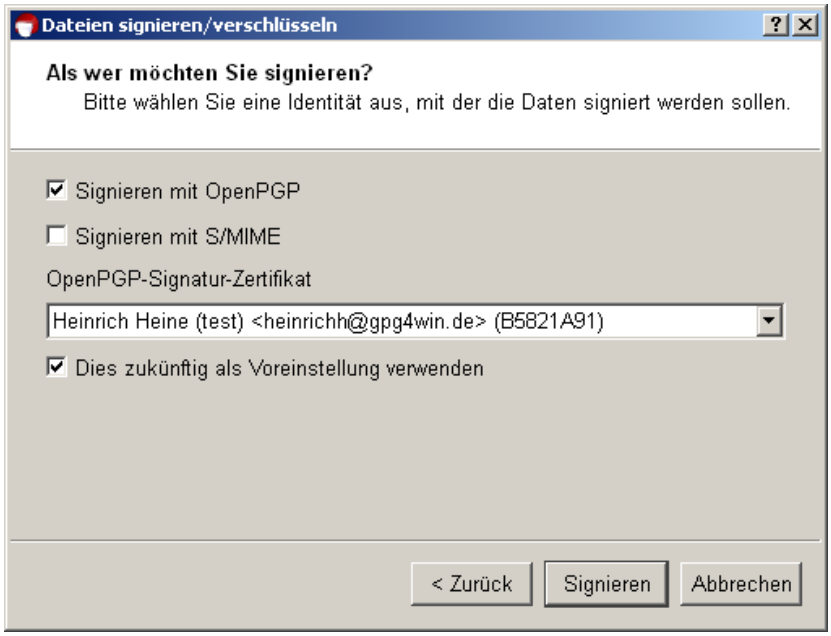
Dort wählen Sie *Signieren und verschlüsseln* aus.

Selektieren Sie im erscheinenden Fenster die Option *Signieren*:



Klicken Sie anschließend auf [*Weiter*].

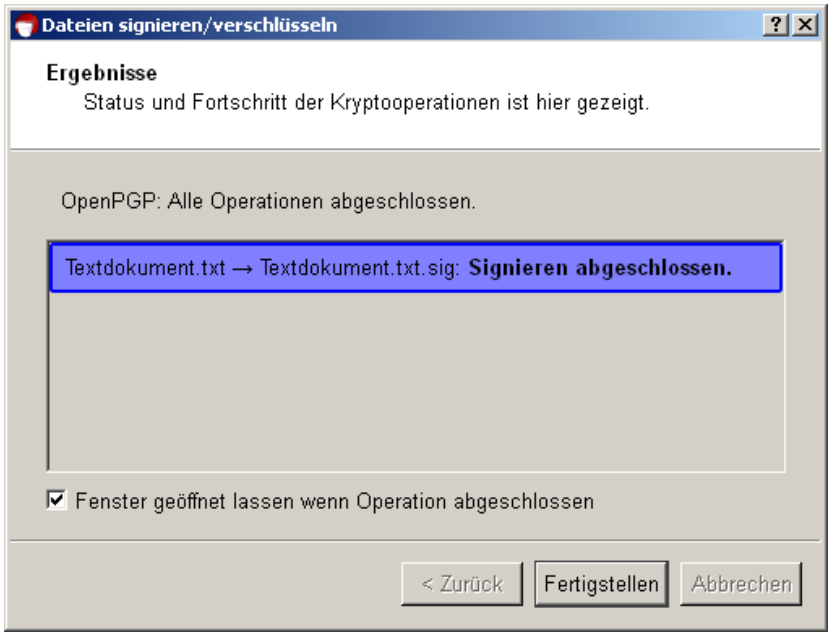
Im folgenden Dialog wählen Sie – sofern nicht schon vorausgewählt – Ihr privates (OpenPGP oder S/MIME) Zertifikat aus, mit dem Sie die Datei signieren möchten.



Bestätigen Sie Ihre Auswahl mit [*Signieren*].

Sie müssen nun Ihre Passphrase in den aufkommenden Pinentry-Dialog eingeben.

Nach erfolgreicher Signierung erhalten Sie folgendes Fenster:



Sie haben damit Ihre Datei erfolgreich signiert.

Abhängig davon, ob Sie OpenPGP oder S/MIME zum Signieren genutzt haben, erhalten Sie als Ergebnis eine Datei mit der Endung `.sig` (bei OpenPGP) oder `.p7s` (bei S/MIME).

Beim Signieren einer Datei wird stets eine „abgetrennte“ (separate) Signatur verwendet. Dies bedeutet, dass Ihre zu signierende Datei unverändert bleibt und eine zweite Datei mit der eigentlichen Signatur erzeugt wird. Um die Signatur später zu prüfen, sind beide Dateien notwendig.

Folgendes Beispiel zeigt noch einmal, welche neue Datei Sie erhalten, wenn Sie Ihre ausgewählte Datei (hier `<dateiname>.txt`) mit OpenPGP bzw. S/MIME signieren:

OpenPGP:

`<dateiname>.txt` → `<dateiname>.txt.sig`

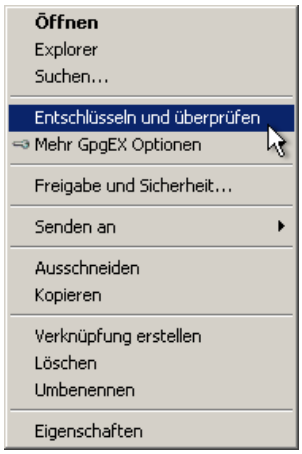
S/MIME:

`<dateiname>.txt` → `<dateiname>.txt.p7s`

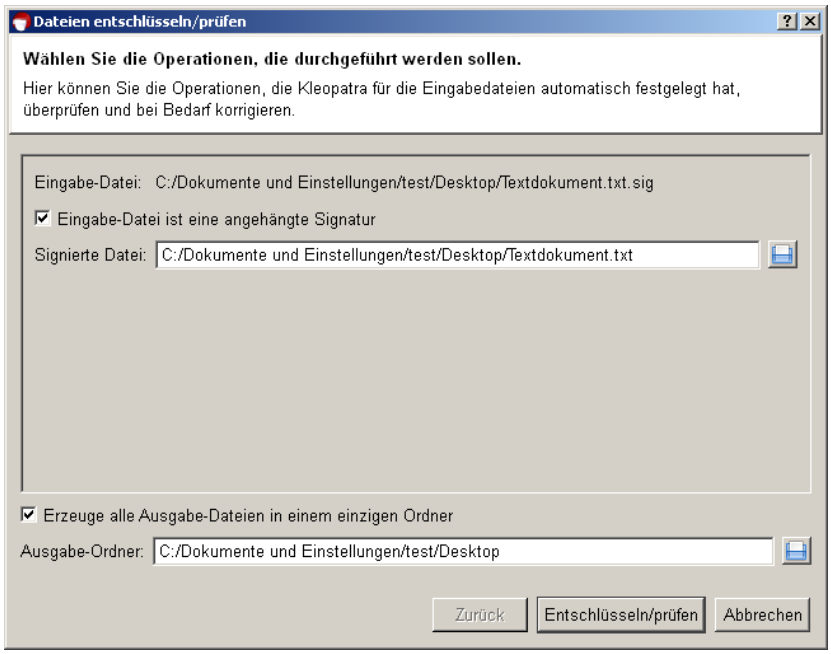
Signatur prüfen

Prüfen Sie nun, ob die eben signierte Datei integer (korrekt) ist!

Zum Überprüfen der Unverändertheit (Integrität) und der Authentizität müssen die Signatur-Datei und die signierte Datei (Originaldatei) in dem selben Dateiordner liegen. Selektieren Sie die Signatur-Datei – also die mit der Endung `.sig` oder `.p7s` – und wählen Sie aus dem Kontextmenü des Windows Explorer den Eintrag *Entschlüsseln und überprüfen*:



Daraufhin erhalten Sie folgendes Fenster:



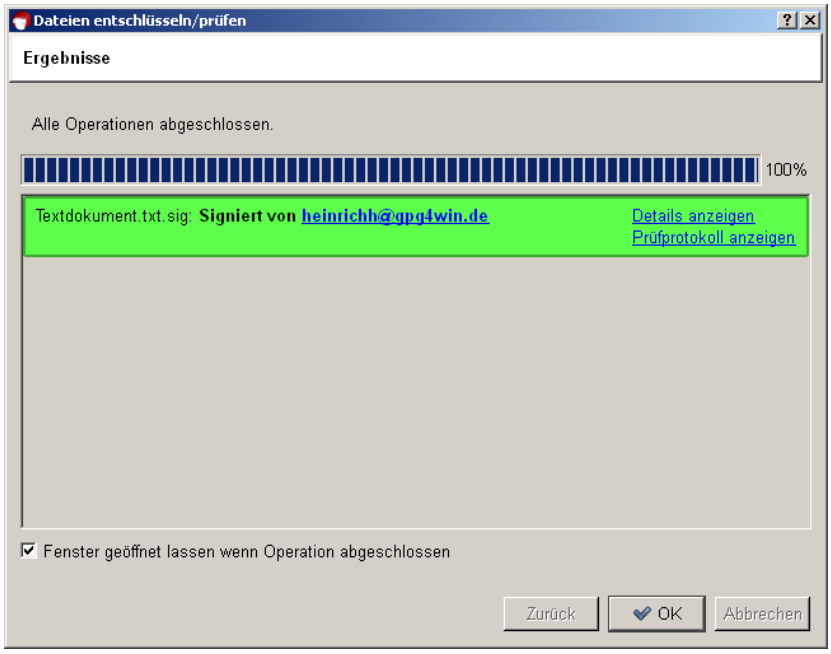
Kleopatra zeigt unter *Eingabe-Datei* den vollständigen Pfad zur ausgewählten Signatur-Datei an.

Die Option *Eingabe-Datei ist eine angehängte Signatur* ist aktiviert, da Sie ja Ihre Originaldatei (hier: *signierte Datei*) mit der Eingabe-Datei signiert haben. Kleopatra findet automatisch die zugehörige signierte Originaldatei in demselben Datei-Ordner.

Automatisch ist auch der *Ausgabe-Ordner* auf den gleichen Pfad ausgewählt. Der wird aber erst relevant wenn Sie mehr als eine Datei gleichzeitig verarbeiten.

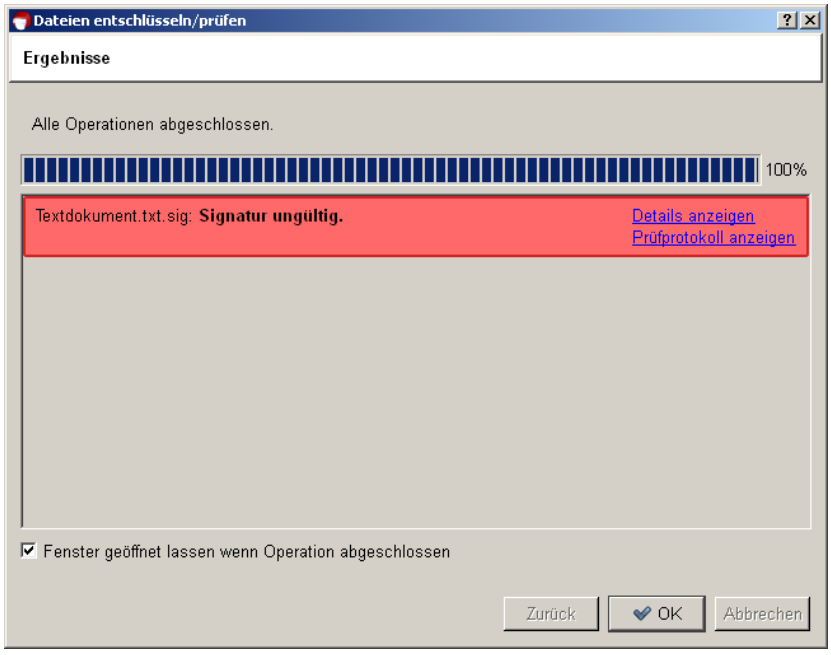
Bestätigen Sie die gegebenen Operationen mit [*Entschlüsseln/prüfen*].

Nach erfolgreicher Überprüfung der Signatur erhalten Sie folgendes Fenster:



Das Ergebnis zeigt, dass die Signatur korrekt ist – also die Datei integer ist und somit **nicht** verändert wurde.

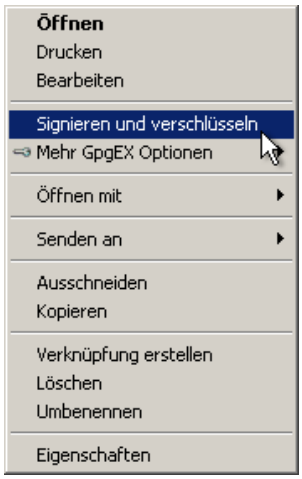
Selbst wenn nur ein Zeichen in der Originaldatei hinzugefügt, gelöscht oder geändert wurde, wird die Signatur als gebrochen angezeigt (Kleopatra stellt das Ergebnis als rote Warnung dar):



18.2 Dateien verschlüsseln und entschlüsseln

Genauso wie E-Mails lassen sich Dateien nicht nur signieren, sondern auch verschlüsseln. Das sollten Sie im folgenden Abschnitt mit GpgEX und Kleopatra einmal durchspielen.

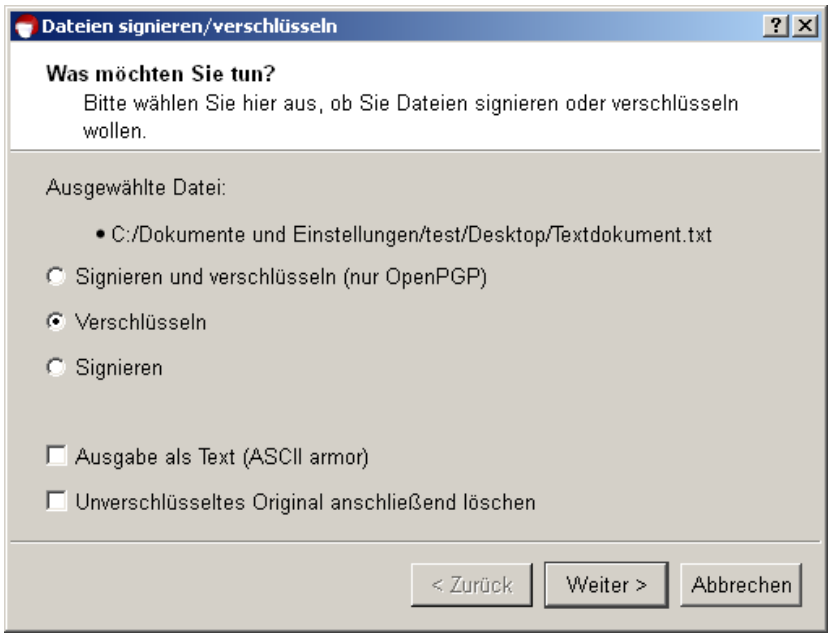
Selektieren Sie eine (oder mehrere) Datei(en) und öffnen Sie mit der rechten Maustaste das Kontextmenü:



Wählen Sie hier *Signieren und verschlüsseln* aus.

Sie erhalten den Dialog, den Sie vom Signieren einer Datei (vgl. Abschnitt 18.1) schon kennen.

Entscheiden Sie sich im oberen Feld für die Option *Nur Verschlüsseln*:



Die Verschlüsselungseinstellungen sollten Sie nur bei Bedarf umstellen:

Ausgabe als Text (ASCII-geschützt / ASCII armor): Bei Aktivierung dieser Option erhalten Sie die verschlüsselte Datei mit der Dateiendung `.asc` (OpenPGP) bzw. `.pem` (S/MIME). Diese Dateitypen sind mit jedem Texteditor lesbar – Sie würden dort den Buchstaben- und Ziffernsalat sehen, den Sie schon kennen.

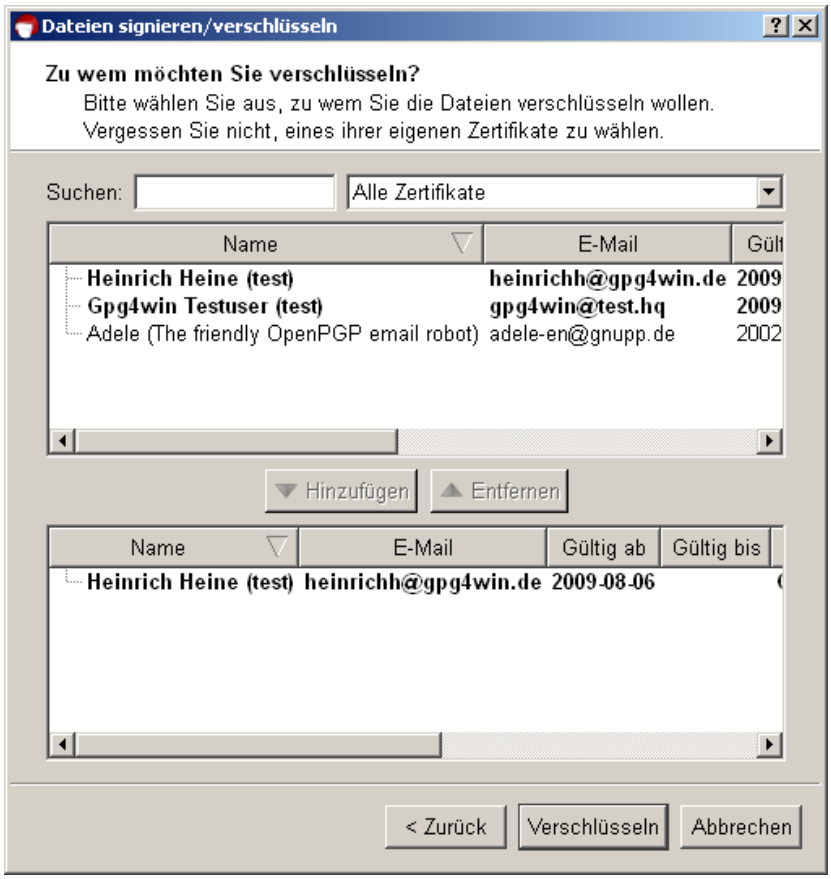
Ist diese Option nicht ausgewählt (Voreinstellung), so wird eine verschlüsselte Datei mit der Endung `.gpg` (OpenPGP) bzw. `.p7m` (S/MIME) angelegt. Diese Dateien sind Binärdateien – eine Betrachtung im Texteditor ist also sinnlos.

Was Sie hier benutzen ist eigentlich gleichgültig; Gpg4win kommt mit beiden Arten klar.

Unverschlüsseltes Original anschließend löschen: Ist diese Option aktiviert, wird Ihre ausgewählte Originaldatei nach dem Verschlüsseln gelöscht.

Klicken Sie auf [*Weiter*].

An wen soll die Datei verschlüsselt werden? Wählen Sie im folgenden Dialog einen oder mehrere Empfängerzertifikate aus:

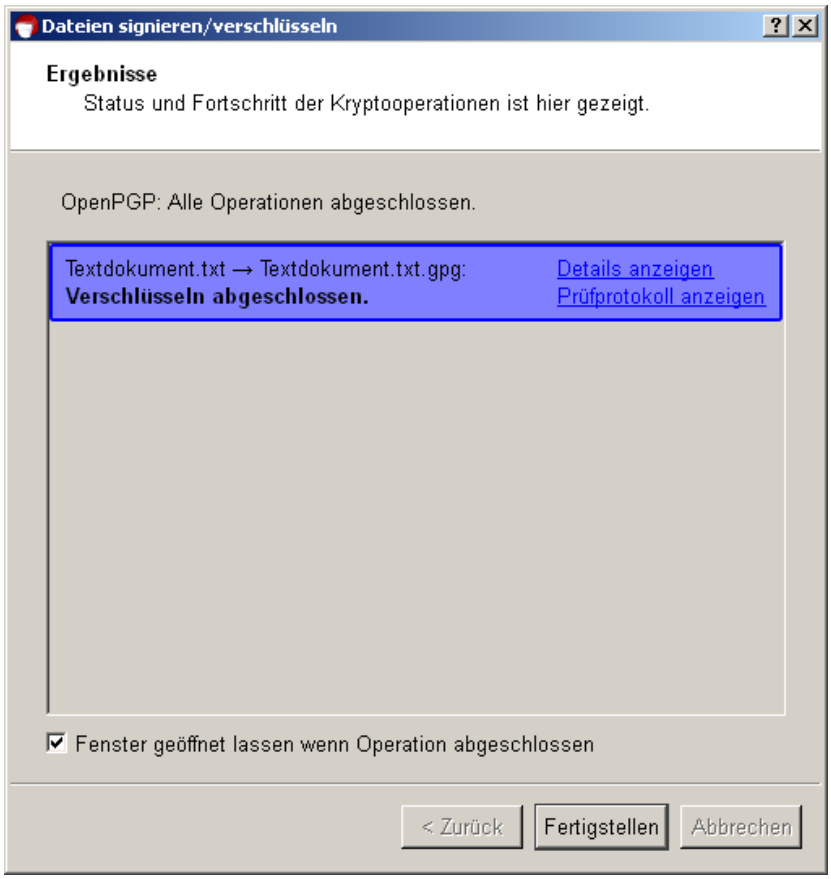


Zum Auswählen selektierten Sie im oberen Teil die gewünschten Zertifikate und drücken Sie auf [*Hinzufügen*]. Sie bekommen alle ausgewählten Zertifikate im unteren Dialogteil zur Kontrolle noch einmal angezeigt.

Abhängig vom gewählten Empfänger-Zertifikat und dessen Typ (OpenPGP oder S/MIME) wird Ihre Datei anschließend mit Hilfe von OpenPGP und / oder S/MIME verschlüsselt. Haben Sie also ein OpenPGP-Zertifikat **und** ein S/MIME-Zertifikat ausgewählt, werden Sie zwei verschlüsselte Dateien erhalten. Die möglichen Dateitypen der verschlüsselten Dateien finden Sie auf der nächsten Seite.

Klicken Sie nun auf [*Verschlüsseln*], um Ihre Datei zu verschlüsseln.

Nach erfolgreicher Verschlüsselung sollte Ihr Ergebnisfenster etwa so aussehen:



Das war's! Sie haben Ihre Datei erfolgreich verschlüsselt!

Wie beim Signieren einer Datei hängt das Ergebnis von der gewählten Verschlüsselungsmethode (OpenPGP oder S/MIME) ab.

Beim Verschlüsseln Ihrer Originaldatei (hier `<dateiname>.txt`) sind insgesamt vier Dateitypen als Ergebnis möglich:

OpenPGP:

`<dateiname>.txt` → `<dateiname>.txt.gpg`
`<dateiname>.txt` → `<dateiname>.txt.asc` (bei Ausgabe als Text/ASCII-armor)

S/MIME:

`<dateiname>.txt` → `<dateiname>.txt.p7m`
`<dateiname>.txt` → `<dateiname>.txt.pem` (bei Ausgabe als Text/ASCII-armor)

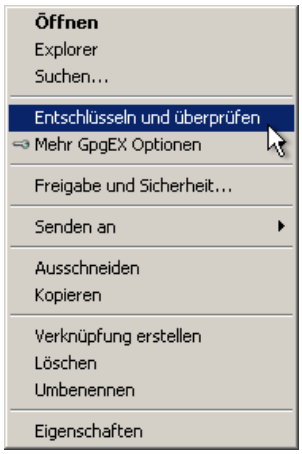
Eine dieser vier möglichen verschlüsselten Ergebnisdateien geben Sie nun an Ihren ausgewählten Empfänger weiter. Anders als beim Signieren einer Datei wird die unverschlüsselte Originaldatei natürlich **nicht** weitergegeben.

Datei entschlüsseln

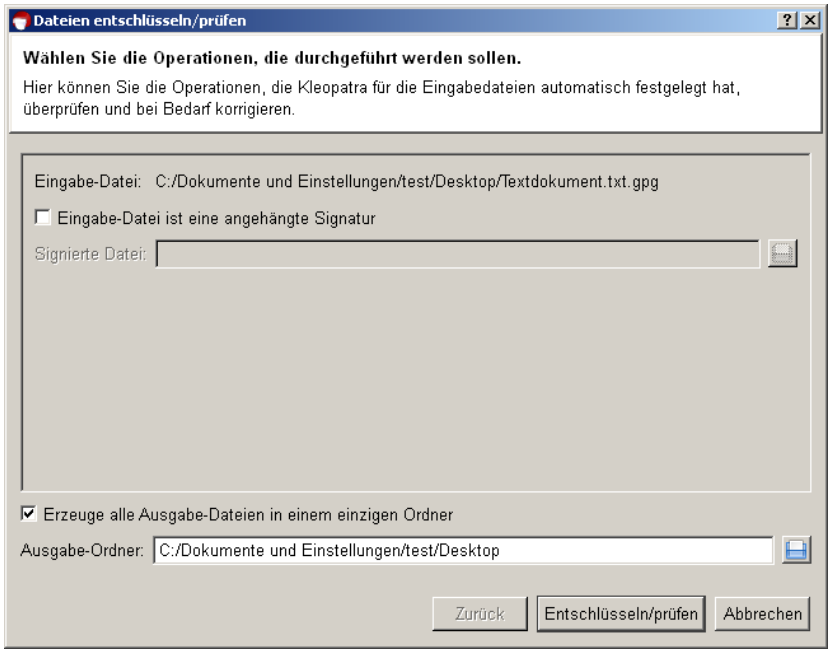
Nun kann die zuvor verschlüsselte Datei zum Testen einmal entschlüsseln werden.

Dazu sollten Sie vorher beim Verschlüsseln auch an Ihr eigenes Zertifikat verschlüsselt haben – andernfalls können Sie die Datei nicht mit Ihrer Passphrase entschlüsseln (vgl. Kapitel 11).

Selektieren Sie die verschlüsselte Datei – also die mit der Endung `.gpg`, `.asc`, `.p7m` oder `.pem` – und wählen Sie aus dem Kontextmenü des Windows Explorer den Eintrag *Entschlüsseln und überprüfen*:



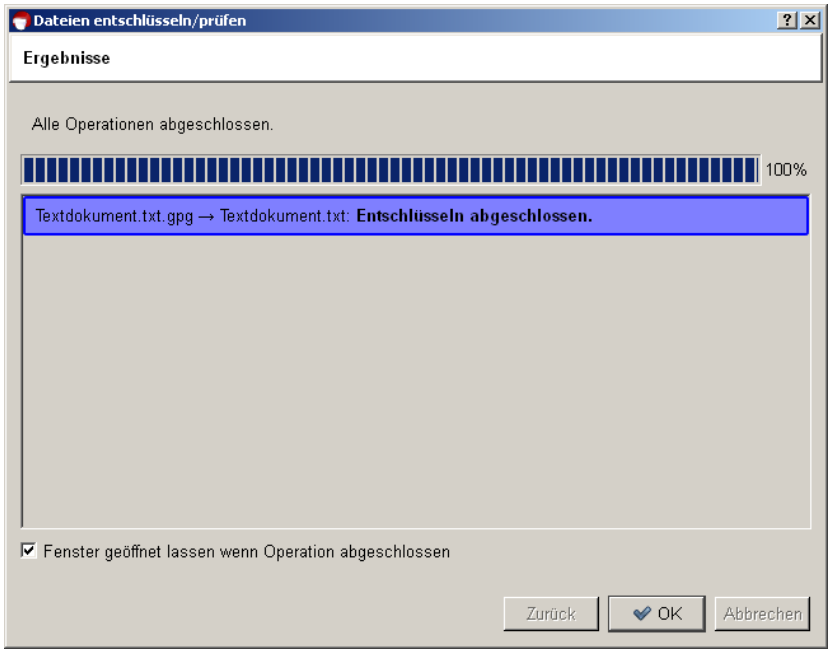
Im folgenden Entschlüsselungsdialog können Sie, bei Bedarf, noch den Ausgabe-Ordner verändern.



Klicken Sie auf [*Entschlüsseln/prüfen*].

Geben Sie anschließend Ihre Passphrase ein.

Das Ergebnis zeigt, dass die Entschlüsselung erfolgreich war:



Sie sollten nun die entschlüsselte Datei problemlos lesen oder mit einem entsprechenden Programm verwenden können.

Kurz zusammengefasst

Sie haben gelernt, wie Sie mit GpgEX:

- Dateien signieren,
- signierte Dateien prüfen,
- Dateien verschlüsseln und
- verschlüsselte Dateien entschlüsseln

können.

Gleichzeitig signieren und verschlüsseln

Ihnen ist diese Option bestimmt schon in den entsprechenden Dialogen aufgefallen. Wählen Sie sie aus, dann kombiniert GpgEX beide Aufgaben in einem Schritt.

Beachten Sie, dass immer *zuerst signiert* erst danach verschlüsselt wird.

Die Signatur wird also immer als geheim mitverschlüsselt. Sie kann nur von denjenigen gesehen und geprüft werden, die die Datei erfolgreich entschlüsseln konnten.

Möchten Sie Dateien signieren *und* verschlüsseln ist das derzeit nur mit OpenPGP möglich.

19 Im- und Export eines geheimen Schlüssels

In den Kapiteln 6 und 8 wurde der Im- und Export von Zertifikaten erläutert. Sie haben Ihr eigenes Zertifikat exportiert, um es zu veröffentlichen, und das Zertifikat Ihres Korrespondenzpartners importiert und am „Schlüsselbund befestigt“.

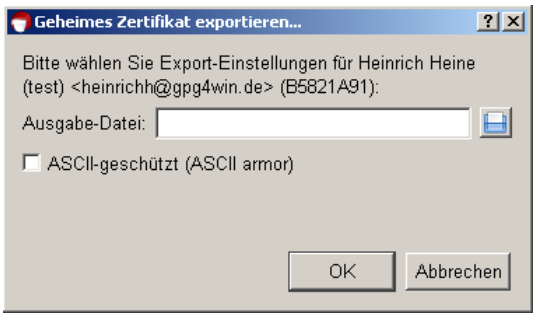
Dabei ging es stets um den **öffentlichen** Schlüssel. Es gibt aber auch hin und wieder die Notwendigkeit, einen **geheimen** Schlüssel zu im- oder exportieren. Wenn Sie z.B. ein bereits vorhandenes (OpenPGP oder S/MIME) Schlüsselpaar mit Gpg4win weiterbenutzen wollen, müssen Sie es importieren. Oder wenn Sie Gpg4win von einem anderen Rechner aus benutzen wollen, muss ebenfalls zunächst das gesamte Schlüsselpaar dorthin transferiert werden – der öffentliche und der private Schlüssel.

19.1 Export

Immer wenn Sie einen geheimen Schlüssel auf einen anderen Rechner transferieren oder auf einer anderen Festplattenpartition bzw. einem Sicherungsmedium speichern wollen, müssen Sie mit Kleopatra eine Sicherungskopie erstellen.

Solch eine Sicherungskopie haben Sie evtl. schon einmal am Ende Ihrer OpenPGP-Zertifikatserzeugung angelegt. Da Ihr OpenPGP-Zertifikat aber inzwischen weitere Beglaubigungen haben kann, sollten Sie die Sicherung erneut durchführen.

Öffnen Sie Kleopatra. Selektieren Sie Ihr eigenes Zertifikat und klicken Sie auf *Datei→Geheimen Schlüssel exportieren.*



Wählen Sie den Pfad und den Dateinamen der Ausgabedatei. Der Dateityp wird automatisch gesetzt. Abhängig davon ob Sie einen geheimen OpenPGP- oder S/MIME-Schlüssel exportieren wollen, ist standardmäßig die Dateiendung `.gpg` (OpenPGP) oder `.p12` (S/MIME) ausgewählt. Bei diesen Dateien handelt es sich um Binärdateien, die Ihr Zertifikat (inkl. geheimen Schlüssel) verschlüsselt enthalten – eine Betrachtung im Texteditor ist hier also sinnlos, aber möglich (probieren Sie es aus).

Bei Aktivierung der Option *ASCII-geschützt (ASCII armor)* erhalten Sie die Dateiendung `.asc` (OpenPGP) bzw. `.pem` (S/MIME). Auch diese beiden Dateitypen sind mit jedem Texteditor lesbar – Sie würden dort den üblichen Buchstaben- und Ziffernsalat sehen. Ob Sie diese Option nutzen oder nicht, ist eigentlich gleichgültig; Gpg4win kommt mit beiden Varianten klar.

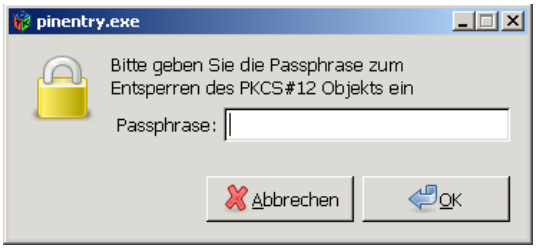
Beide Schlüsselteile – der öffentliche und der geheime – werden von Kleopatra in **einem** einzigen geheimen Zertifikat abgespeichert.

Achtung: Behandeln Sie diese Datei sehr sorgfältig. Sie enthält Ihren geheimen Schlüssel und damit eine sehr sicherheitskritische Informationen!

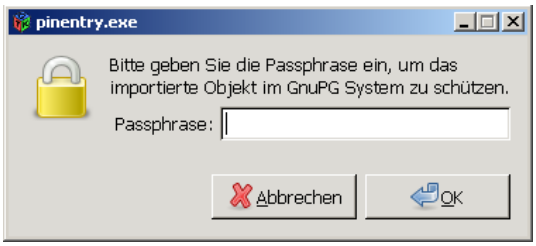
19.2 Import

Zum Importieren Ihres vorher exportierten geheimen Zertifikats in Kleopatra, gehen Sie so vor, wie Sie es beim Importieren von öffentlichen (fremden) Zertifikaten gewohnt sind (vgl. Kapitel 8):

Klicken Sie auf *Datei→Zertifikat importieren...* und wählen Sie die zu importierende Datei aus. Handelt es sich um eine PKCS12-Datei (z.B. vom Typ `.p12`), so werden Sie zunächst nach der Passphrase zum Entsperren des privaten Schlüssels gefragt:



Setzen Sie nun eine (ggf. neue) Passphrase, die nach dem Importvorgang Ihrem privaten Schlüssel zugeordnet werden soll:



Wiederholen Sie Ihre Passphrase-Eingabe. Sollte Ihre Passphrase zu kurz sein oder nur aus Buchstaben bestehen, werden Sie entsprechend gewarnt.

Nach dem erfolgreichen Importieren sehen Sie ein Informationsfenster, das Ihnen die Ergebnisse des Importvorgangs auflistet; hier am Beispiel eines geheimen OpenPGP-Zertifikats:

Kleopatra hat damit sowohl den geheimen als auch den öffentlichen Schlüssel aus der Sicherungsdatei importiert. Ihr Zertifikat ist damit unter „Meine Zertifikate“ in der Zertifikatsverwaltung von Kleopatra sichtbar.

Achtung: Löschen Sie danach unbedingt die oben erstellte Sicherungskopie Ihres geheimen Schlüssels von Ihrer Festplatte und sichern Sie vorher diese wichtige Datei möglichst irgendwo auf einem externen Medium. Denken Sie daran die gelöschte Datei aus Ihrem „Papierkorb“ zu entfernen. Andernfalls stellt diese Datei ein großes Sicherheitsrisiko für Ihre geheime E-Mail-Verschlüsselung dar!

Anmerkung: Es kann in einigen Fällen vorkommen, dass Sie einen importierten Schlüssel nicht direkt benutzen können. Dies äußert sich darin, dass Sie die richtige Passphrase eingeben, dieser aber nicht akzeptiert wird. Das kommt daher, dass einige Versionen von PGP intern den IDEA Algorithmus verwenden. Dieser kann von GnuPG aus rechtlichen Gründen nicht unterstützt werden. Um das Problem zu beheben, ändern Sie in PGP einfach die Passphrase und exportieren / importieren Sie den Schlüssel erneut. Sollte dies auch nicht funktionieren, so setzen Sie die Passphrase in PGP auf „leer“; d.h. auf keinen Schutz und exportieren / importieren Sie wieder – In diesem Fall müssen Sie unbedingt sicherstellen, sowohl die **Datei sicher** zu **löschen** als auch in PGP und in Gpg4win danach wieder eine echte **Passphrase zu setzen.**

Herzlichen Glückwunsch! Sie haben damit erfolgreich Ihr Schlüsselpaar exportiert und wieder importiert.

20 Systemweite Konfigurationen und Vorbelegungen für S/MIME

S/MIME

Im Rahmen von Softwareverteilung oder sonstigen Umgebungen, in denen viele Anwender auf einem Rechner arbeiten, ist es sinnvoll einige systemweite Vorgaben und Vorbelegungen für Gpg4win einzurichten.

Das betrifft vor allem S/MIME, denn bei streng vorgegebenen Vertrauensketten macht es Sinn, dass die Anwender die Informationen dazu miteinander teilen.

Einige typische systemweite Einstellungen sind:

- Vertrauenswürdige Wurzelzertifikate

  Um zu vermeiden, dass jeder Anwender selbst die notwendigen Wurzelzertifikate suchen und installieren sowie deren Vertrauenswürdigkeit prüfen und bestätigen (beglaubigen) muss (vgl. Abschnitt 22.7), ist eine systemweite Vorbelegung der wichtigsten Wurzelzertifikate sinnvoll.

  Dafür sind folgende Schritte durchzuführen:

  1. Die Wurzelzertifikate ablegen wie unter Abschnitt 22.3 beschrieben.
  2. Die vertrauenswürdigen Wurzelzertifikate definieren wie unter Abschnitt 22.6 beschrieben.

- Direkt verfügbare CA-Zertifikate

  Um den Anwendern zusätzlich die Mühe zu ersparen, die Zertifikate der Beglaubigungsinstanzen (Certificate Authorities, CAs) zu suchen und zu importieren, ist auch hier eine systemweite Vorbelegung der wichtigsten CA-Zertifikate sinnvoll.

  Folgen Sie dazu der Beschreibung unter Abschnitt 22.4.

- Proxy für Zertifikatsserver-Suche Es kommt vor, dass interne Netzwerke keine direkten Verbindungen der einzelnen Rechner nach aussen zulassen, sondern einen sogenannten Proxy vorsehen.

  Ist dies in Ihrem Netzwerk auch für die bei GnuPG bzw. S/MIME wichtigen LDAP-Abfragen der Fall, so führen Sie folgende Schritte durch:

  1. Stellen Sie X.509-Zertifikatsserver-Suchen auf Ihren Proxy wie unter Abschnitt 22.5 ein.
  2. Stellen Sie Sperrlisten-Suchen auf Ihren Proxy ein, in dem Sie einen Eintrag wie z.B.:
     `http-proxy http://proxy.mydomain.example:8080`
     (ggf. analog für LDAP) als Administrator in die Datei
     `C:\Dokumente und Einstellungen\All Users\GNU\etc\dirmngr\dirmngr.conf`
     eintragen.
  3. Starten Sie den DirMngr neu (siehe Abschnitt 21.6).

21 Bekannte Probleme und was man tun kann

21.1 GpgOL Menüs und Dialoge nicht mehr in Outlook zu finden

Es kann vorkommen, dass trotz einer Aktualisierung von Gpg4win die von GpgOL zu Outlook hinzugefügten Menüs und Dialoge nicht mehr zu finden sind.

Das kann dann vorkommen, wenn ein technisches Problem auftrat und Outlook aus diesem Grund die GpgOL-Komponente deaktiviert.

Reaktivieren Sie GpgOL über das Outlook-Menü:
Outlook2007: *?→Deaktivierte Elemente*
Outlook2003: *?→Info→Deaktivierte Elemente*

Um unter Outlook2003 GpgOL manuell zu (de-)aktivieren nutzen Sie den Add-In-Manager von Outlook: *Extras→Optionen→Weitere→Erweiterte Optionen...→Add-In-Manger...*

21.2 GpgOL-Schaltflächen sind in Outlook2003 nicht in der Symbolleiste

Wenn bereits viele Schaltflächen in der Symbolleiste des Nachrichtenfensters vorhanden sind, so stellt Outlook2003 die Signieren- / Verschlüsseln-Icons von GpgOL nicht unbedingt sofort sichtbar dar.

Sie können diese Schaltflächen aber anzeigen lassen, indem Sie in der Symbolleiste auf das kleine Icon mit dem Pfeil nach unten klicken (*Optionen für Symbolleiste*): Sie erhalten eine Übersicht aller nicht angezeigten Schaltflächen. Ein Klick auf eines dieser Einträge verschiebt ihn in den sichtbaren Teil der Symbolleiste.

21.3 GpgOL-Schaltflächen sind in Outlook2007 unter „Add-Ins“

Mit Outlook2007 wurde die sogenannte „Ribbon“-Oberfläche eingeführt. Diese Multifunktionsleiste im Outlook-Nachrichtenfenster besitzt verschiedene Registerkarten. Die GpgOL-Schaltflächen (für Verschlüsseln, Signieren etc.) sind unter der Registerkarte „Add-Ins“ eingeordnet; so wie alle Schaltflächen von Erweiterungen durch Outlook dort angelegt werden. Eine Integration der GpgOL-Schaltflächen z.B. unter „Nachrichten“ ist nicht möglich.

Was Sie machen können, ist Ihre *Symbolleiste für den Schnellzugriff* anzupassen und die Symbolleistenbefehle der Add-In-Registerkarte aufnehmen.

21.4 Fehler beim Start von GpgOL

Haben Sie Gpg4win (und damit die Programmkomponente GpgOL) auf einem Laufwerk installiert, anschließend wieder deinstalliert und unter einem anderen Laufwerk erneut installiert? Dann kann es sein, dass Outlook weiterhin den GpgOL-Pfad auf dem ersten (alten) Laufwerk sucht.

Dabei wird beim Start von Outlook die Programmerweiterung GpgOL nicht mehr gestartet und folgende Fehlermeldung erscheint:

Die Erweiterung '<alter-Pfad-zu-gpgol.dll>' konnte nicht installiert oder geladen werden. Das Problem kann u.U. durch das Benutzen von 'Erkennen und Reparieren' in der Hilfe behoben werden.

Lösen können Sie dieses Problem, in dem Sie den Outlook-internen (zwischengespeicherten) Programmerweiterungs-Pfad zurücksetzen. Löschen Sie dazu bitte folgende Datei:
`%APPDATA%\Lokale Einstellungen\Anwendungsdaten\Microsoft\Outlook\extend.dat`

Dabei sollte Outlook natürlich nicht laufen. Anschließend starten Sie Outlook erneut. Outlook und GpgOL sollte nun problemlos starten.

Beachten Sie bitte auch, dass eine Installation von Gpg4win auf einem (mit dem Befehl `subst` simulierten) **virtuellen Laufwerk** nicht möglich ist. Diese virtuellen Laufwerke sind nur lokal für den aktuellen Benutzer nutzbar. Systemdienste (wie der DirMngr) sehen diese Laufwerke nicht. Der Installationspfad ist damit ungültig - die Installation stoppt mit einem Fehler in der Art `error:StartService: ec=3`. Installieren Sie bitte Gpg4win auf einem systemweit verfügbaren Laufwerk.

21.5 GpgOL überprüft keine InlinePGP E-Mails von „CryptoEx“

Um signierte bzw. verschlüsselte InlinePGP E-Mails, die von Outlook-Programmerweiterung „CryptoEx“ versandt wurden, zu prüfen bzw. zu entschlüsseln, muss in den GpgOL-Optionen die S/MIME-Unterstützung eingeschaltet sein.

Aktivieren Sie dazu in Outlook unter *Extras→Optionen→GpgOL* die Option:
S/MIME Unterstützung einschalten.

21.6 Keine S/MIME Operationen mehr möglich (Systemdienst „DirMngr“ läuft nicht)

S/MIME

Der „Directory Manager“ (DirMngr) ist ein über Gpg4win installierter Dienst, der die Zugriffe auf Zertifikatsserver verwaltet. Eine Aufgabe des DirMngr ist das Laden von Sperrlisten (CRLs) für S/MIME Zertifikate.

Es kann vorkommen, dass die S/MIME Operationen (Signatur, Prüfung, Ver- oder Entschlüsselung) nicht durchgeführt werden können, weil DirMngr nicht verfügbar ist. In der Voreinstellung von Gpg4win ist es zwingend notwendig, dass DirMngr die Sperrliste prüft – geschieht das nicht, darf die jeweilige Operation nicht ausgeführt werden, da möglicherweise ein kompromittiertes Zertifikat genutzt wird.

Abhilfe schaffe ein Neustart des DirMngr durch den Systemadministrator. Dies erfolgt über *Systemsteuerung→Verwaltung→Dienste*. In der Liste finden Sie DirMngr – über das Kontextmenü kann der Dienst neu gestartet werden.

22 Wo finde ich die Dateien und Einstellungen von Gpg4win?

22.1 Persönliche Einstellungen der Anwender

Die persönlichen Einstellungen für jeden Anwender befinden sich im Dateiordner:
`%APPDATA%\gnupg`
Oft entspricht das dem Dateiordner:
`C:\Dokumente und Einstellungen\<name>\Anwendungsdaten\gnupg\`

Beachten Sie, dass es sich um einen versteckten Dateiordner handelt. Um es sichtbar zu schalten, müssen Sie im Explorer über das Menü *Extras→Ordneroptionen* im Reiter *Ansicht* die Option *Alle Dateien und Ordner anzeigen* unter der Gruppe *Versteckte Dateien und Ordner* aktivieren.

In diesem Dateiordner befinden sich sämtliche persönlichen GnuPG-Daten, also die privaten Schlüssel, Zertifikate, Vertrauensstellungen und Konfigurationen. Bei einer Deinstallation von Gpg4win wird dieser Ordner *nicht* gelöscht. Denken Sie daran, sich regelmäßig Sicherheitskopien von diesem Ordner anzulegen.

22.2 Zwischengespeicherte Sperrlisten

S/MIME

Der systemweite Dienst DirMngr (Directory Manager) prüft unter anderem, ob ein X.509-Zertifikat gesperrt ist und daher nicht verwendet werden darf. Dafür werden Sperrlisten (CRLs) von den Ausgabestellen der Zertifikate („Trust-Center") abgeholt und für die Dauer ihrer Gültigkeit zwischengespeichert.

Abgelegt werden diese Sperrlisten unter:
`C:\Dokumente und Einstellungen\LocalService\Lokale Einstellungen\Anwendungsdaten\GNU\cache\dirmngr\crls.d\`

Hierbei handelt es sich um *geschützte* Dateien, die standardmäßig vom Explorer nicht angezeigt werden. Sollten Sie dennoch die Anzeige dieser Dateien wünschen, deaktivieren Sie die Option *Geschützte Systemdateien ausblenden* in den *Ansicht*-Einstellungen des Windows Explorer.

In diesem Dateiordner sollten keine Änderungen vorgenommen werden.

22.3 Vertrauenswürdige Wurzelzertifikate von DirMngr

S/MIME

Für eine vollständige Prüfung von X.509-Zertifikaten muss auch den Wurzelzertifikaten vertraut werden, mit deren Hilfe die Sperrlisten signiert wurden.

Die Wurzelzertifikate, denen der DirMngr bei den Prüfungen vertrauen soll, müssen im folgenden Dateiordner abgelegt werden:

```
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\
etc\dirmngr\trusted-certs\
```

Wichtig: Die entsprechenden Wurzelzertifikate müssen als Dateien im Dateiformat DER mit Dateinamens-Erweiterung `.crt` oder `.der` im o.g. Dateiordners vorliegen.

Der DirMngr läuft als systemweiter Dienst und muss nach Änderungen im „trusted-certs“-Dateiordner neu gestartet werden. Anschließend sind die dort abgelegten Wurzelzertifikate für alle Anwender als **vertrauenswürdig** gesetzt.

Beachten Sie auch Abschnitt 22.6, um den Wurzelzertifikaten vollständig (systemweit) zu vertrauen.

22.4 Weitere Zertifikate von DirMngr

S/MIME

Wenn vor einer Krypto-Operation die X.509-Zertifikatskette zu prüfen ist, ist somit auch das jeweiliges Zertifikat der Beglaubigungsinstanz („Certificate Authority“, CA) zu prüfen.

Für eine direkte Verfügbarkeit können CA-Zertifikate in diesem (systemweiten) Dateiordner abgelegt werden:
```
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\
lib\dirmngr\extra-certs\
```

Zertifikate, die nicht hier oder bei den Anwendern vorliegen, müssen entweder automatisch von X.509-Zertifikatsservern geladen werden. Alternativ können Zertifikate auch immer manuell importiert werden.

Es ist also sinnvoll im Rahmen von systemweiten Vorgaben hier die wichtigsten CA-Zertifikate abzulegen.

22.5 Konfiguration zur Verwendung externer X.509-Zertifikatsserver

S/MIME

GnuPG kann so konfiguriert werden, dass bei Bedarf fehlende X.509-Zertifikate oder Sperrlisten auf externen X.509-Zertifikatsserver gesucht werden.

Der Systemdienst DirMngr verwendet dafür die Liste der Dienste, die in der Datei
```
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\
etc\dirmngr\ldapservers.conf
```
angegeben sind.

Sind im internen Netz die Zugänge zu externen Zertifikatsservern per LDAP gesperrt, so kann man in dieser Datei einen Proxy-Dienst für entsprechende Durchleitung konfigurieren, wie folgende Zeile im Beispiel illustriert:

```
proxy.mydomain.example:389:::O=myorg,C=de
```

Die genaue Syntax für die Einträge lautet:

```
HOSTNAME:PORT:USERNAME:PASSWORD:BASE_DN
```

22.6 Systemweite vertrauenswürdige Wurzelzertifikate

S/MIME

Die systemweit als vertrauenswürdig vorbelegten Wurzelzertifikate werden definiert in der Datei `C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\etc\gnupg\trustlist.txt`

Um ein Wurzelzertifikat als vertrauenswürdig zu markieren, muss der entsprechende Fingerabdruck des Zertifikats gefolgt von einem Leerzeichen und einem großen `S` in die o.g. Datei eingetragen werden. Ein Zertifikat wird explizit als nicht vertrauenswürdig markiert, wenn die Zeile mit dem Präfix „`!`“ beginnt. Sie können hier auch mehrere Wurzelzertifikate eintragen. Zu beachten ist dann, dass jeder Fingerabdruck in einer neuen Zeile steht.

Wichtig: Abschließend (am Ende der Datei) muss eine Leerzeile erfolgen.

Ein Beispiel:

```
# CN=Wurzel ZS 3,O=Intevation GmbH,C=DE
A6935DD34EF3087973C706FC311AA2CCF733765B S

# CN=PCA-1-Verwaltung-02/O=PKI-1-Verwaltung/C=DE
DC:BD:69:25:48:BD:BB:7E:31:6E:BB:80:D3:00:80:35:D4:F8:A6:CD S

# CN=Root-CA/O=Schlapphuete/L=Pullach/C=DE
!14:56:98:D3:FE:9C:CA:5A:31:6E:BC:81:D3:11:4E:00:90:A3:44:C2 S
```

Es kann in einigen Fällen sinnvoll sein, die Kriterien an die Überprüfung der Wurzelzertifikate zu verringern. Sie können dazu hinter `S` eine weitere Flagge `relax` setzen: `<FINGERABDRUCK> S relax`

Wichtig: Die Verwendung von `relax` setzt die Sicherheit herab und muss individuell entschieden werden und sollte nur bei Problemen verwendet werden.

Genauere Details finden Sie in der aktuellen GnuPG-Dokumentation (Punkt „trustlist.txt“):
`http://www.gnupg.org/documentation/manuals/gnupg/Agent-Configuration.html`

Die genaue Syntax für die Einträge in die trustlist.txt lautet also:
`[!]<FINGERABDRUCK> S [relax]`
wobei `!` und `relax` optional sind.

Anstelle der Flagge `S` sind noch die Werte `P` und `*` vorgesehen, die für zukünftigen Gebrauch reserviert sind.

Wichtig: Damit Wurzelzertifikate in Kleopatra vollständig als vertrauenswürdig markiert werden (Zertifikat wird blau hinterlegt), müssen die Wurzelzertifikate zusätzlich für den DirMngr abgelegt werden, wie unter Abschnitt 22.3 beschrieben.

22.7 Vertrauenswürdigkeit der Wurzelzertifikate durch Benutzer markieren

Wurzelzertifikate können auch jeweils von den einzelnen Benutzern als vertrauenswürdig markiert werden – eine systemweite Konfiguration (siehe Abschnitt 22.3 und 22.6) ist dann nicht erforderlich.

Öffnen Sie das Kleopatra-Menü *Einstellungen→Kleopatra einrichten* und anschließend die Gruppe *GnuPG-System* und dann *GpgAgent*.
Aktivieren Sie hier die Option *Erlaube Aufrufern Schlüssel als 'vertrauenswürdig' zu markieren*. Dadurch werden Sie beim Gebrauch eines bisher nicht vertrauenswürdig eingestuften Wurzelzertifikats gefragt, ob Sie es nun als vertrauenswürdig einstufen wollen. Beachten Sie, dass der gpg-agent ggf. neu gestartet werden muss (z.B. durch ausloggen und wieder einloggen).

Die von Ihnen als vertrauenswürdig (oder wahlweise explizit als nicht vertrauenswürdig) gekennzeichneten Wurzelzertifikate werden automatisch in folgender Datei gespeichert:
`C:\Dokumente und Einstellungen\<Nutzername>\Anwendungsdaten\gnupg\trustlist.txt`

Für die trustlist.txt gilt die gleiche Syntax wie im Abschnitt 22.6 beschrieben.

23 Probleme in den Gpg4win-Programmen aufspüren

Es kann vorkommen, dass eine der Gpg4win-Programmkomponenten nicht wie erwartet zu funktionieren scheint.

Nicht selten ist dabei eine Besonderheit der Arbeitsumgebung verantwortlich, so dass die Softwareentwickler von Gpg4win das beobachtete Problem gar nicht selbst nachvollziehen können.

Um die Softwareentwickler bei der Problemsuche zu unterstützen oder um auch einmal selbst in die technischen Detail-Abläufe reinzuschnuppern, bieten die Gpg4win-Programme Unterstützung an.

In der Regel muss diese Unterstützung aber erst einmal eingeschaltet werden. Eine der wichtigsten Hilfsmittel sind Logdateien: Dort werden detaillierte Informationen zu den internen technischen Vorgängen festgehalten, wie in einer Logdatei. Ein Softwareentwickler kann ein Problem und die mögliche Lösung oft leicht anhand dieser erkennen, auch wenn das Problem auf den ersten Blick sehr unverständlich und zu umfangreich wirken mag.

Wenn Sie einen Fehler-Bericht an die Softwareentwickler senden wollen, so finden Sie auf dieser Web-Seite einige Hinweise:

`http://www.gpg4win.de/reporting-bugs-de.html`

Logdateien – unter o.g. URL als „Debug-Informationen” bezeichnet – bieten oft wertvolle Hinweise und sollten daher einem Fehlerbericht beigefügt werden.

In diesem Kapitel wird beschrieben, wie Sie Programmablauf-Informationen (darum handelt es sich letztlich bei den Logdateien) zu den einzelnen Gpg4win-Programmen einschalten können.

23.1 Logdatei von Kleopatra einschalten

Die Logdatei von Kleopatra besteht aus vielen Dateien, daher ist der erste Schritt ein Dateiordner für die Logdatei zu erstellen. Denkbar ist z.B.: `C:\TEMP\kleologdir`

Bitte beachten Sie hierbei, dass es hier um Einstellungen des Anwenders, nicht des Systemadministrators geht. Die Einstellungen müssen also für jeden Anwender der eine Logdatei erstellen möchte separat vorgenommen werden und darauf geachtet werden, dass unterschiedliche `kleologdir` Dateiordner verwendet werden.

Der Pfad zu diesem Ordner muss nun in der neuen Umgebungsvariable `KLEOPATRA_LOGDIR` vermerkt werden:

Öffnen Sie dazu die Systemsteuerung, wählen dort *System*, dann den Reiter *Erweitert* und schließlich den Knopf [*Umgebungsvariablen*].

Fügen Sie dort folgende neue **Benutzervariable** ein:

> Name der Variable: `KLEOPATRA_LOGDIR`
>
> Wert der Variable: `C:\TEMP\kleologdir`

Beachten Sie, dass der angegebene Dateiordner existieren muss. Sie können es auch nachträglich erstellen.

Um die Logfunktion wirksam werden zu lassen, muss Kleopatra beendet und neu gestartet werden und der Dateiordner der Logdatei existieren, sowie für Kleopatra beschreibbar sein.

Während Kleopatra verwendet wird, zeichnet es Ablauf-Informationen in die Datei `kleo-log` (Haupt-Logdatei) auf, sowie möglicherweise viele Dateien mit einem Namen nach dem Schema:
`pipe-input-ZEITSTEMPEL-ZUFALLSZEICHEN`

Möglicherweise reichen diese Informationen einem Softwareentwickler nicht, um den Fehler zu erkennen. Er wird Sie dann bitten, eine weitere Umgebungsvariable anzulegen – so wie Sie es schon oben getan haben:

> Name der Variable: `KLEOPATRA_LOGOPTIONS`
>
> Wert der Variable: `all`

Möglicherweise werden die Logdateien sehr schnell sehr groß. Sie sollten diese Logdatei-Aufzeichnung nur einschalten, um ein bestimmtes Fehlverhalten zu provozieren und danach aufzuzeichnen. Anschliessend schalten Sie die Aufzeichnung wieder aus indem Sie die Umgebungsvariable löschen oder den Namen dieser leicht variieren (für späteres leichtes Reaktivieren). Vergessen Sie nicht, die Logdateien zu löschen gerade wenn sie sehr umfangreich geworden sind oder es sich um sehr viele Dateien handelt. Bevor Sie eine neue Aufzeichnung beginnen, ist es ebenfalls sinnvoll die Logdateien zu löschen.

23.2 Logdatei von GpgOL einschalten

Für das Einschalten der Logdatei von GpgOL müssen Sie den Registrierungs-Editor starten. Geben Sie dazu das Kommando `regedit` unter *Start→Ausführen* oder in einer Eingabeaufforderung ein.

Wählen Sie nun aus der Baumstruktur auf der linken Seite den folgenden GpgOL-Schlüssel aus:
`HKEY_CURRENT_USER\Software\GNU\GpgOL`

Auf der rechten Seite sehen Sie nun eine Liste von Einträgen (sogenannte Zeichenfolgen) mit teilweise bereits vordefinierten Werten. Diese Einträge werden nach dem ersten Start von Outlook mit GpgOL angelegt.

Zum Aktivieren der GpgOL-Logdatei führen Sie einen Doppelklick auf den Eintrag `enableDebug` aus und setzte Sie dessen Wert auf `1`.

Als Wert für `logFile` geben Sie nun einen Dateinamen an, wohin die Logdatei geschrieben werden soll, z.B.: `C:\TEMP\gpgol.log`

Starten Sie Outlook neu, um die Aufzeichnung zu starten.

Bedenken Sie, dass diese Datei sehr umfangreich werden kann. Stelle Sie `enableDebug` auf `0`, sobald Sie die GpgOL-Logdatei nicht mehr benötigen.

Sollen Sie sich für fortgeschrittene technische Informationen zu GpgOL interessieren, dann lesen Sie einmal das technische (englischsprachige) Handbuch. Sie finden es in Ihrem Gpg4win-Installationsverzeichnis, in der Regel: `C:\Programme\GNU\GnuPG\share\doc\gpgol\gpgol.pdf`

23.3 Logdatei von DirMngr einschalten

Bei DirMngr handelt es sich um einen systemweiten Dienst und daher ist das Einschalten der Logdatei nur mit Administatorrechten möglich.

Um die Logdatei einzuschalten, öffnen Sie zunächst folgende Konfigurationsdatei:
`C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\etc\dirmngr\dirmngr.conf`

Fügen Sie die folgenden zwei Zeilen in die Konfigurationsdatei hinzu (den Pfad zur Logdatei können Sie natürlich anpassen):

```
debug-all
log-file C:\TEMP\dirmngr.log
```

Starten Sie anschließend den Dienst unter *Systemsteuerung→Verwaltung→Dienste* neu, so dass die geänderte Konfigurationsdatei neu eingelesen wird und die vorgenommene Einstellungen somit wirksam wird.

23.4 Logdatei von GnuPG einschalten

Für folgende GnuPG-Programme können Sie jeweils einzeln das Anlegen einer Logdatei einschalten:

- GPG Agent
- GPG für S/MIME
- GPG für OpenPGP
- Smartcard Daemon

Für diese Programme können Anwender persönliche Konfigurationen vornehmen. Dazu gehört auch das Einstellen einer Protokolldatei für den Programmablauf.

Eingeschaltet wird die jeweilige Logdatei im GnuPG Backend – erreichbar über das Kleopatra Menü *Einstellungen→GnuPG Backend einrichten...*. Für jede der o.g. vier Programme existieren in diesem Konfigurationsfenster zwei Debug-Optionen:

- Option *Setze die Debug-Stufe auf*
  Hier definieren Sie die Ausführlichkeit der aufzuzeichnenden Informationen. Die Debug-Stufe *Guru* ist die höchste Stufe und erzeugt dem entsprechend große Dateien. Schalten Sie daher die Logdateien wieder aus (Debug-Stufe *Keine*), wenn Sie diese nicht mehr benötigen.
- Option *Schreibe im Servermodus Logs auf DATEI*
  Geben Sie hier die Logdatei an, in der alle Debug-Informationen gespeichert werden sollen, z.B.: `C:\TEMP\gpgsm.log`

Starten Sie anschließend Kleopatra neu (ggf. müssen Sie zuvor einen noch laufenden gpg-agent über den Task-Manager beenden – oder aber Sie loggen sich aus und melden sich neu an Ihrem Windows-System an).

24 Warum Gpg4win nicht zu knacken ist . . .

. . ., jedenfalls nicht mit heute bekannten Methoden und sofern die Implementierung der Programme frei von Fehlern ist. Soweit die Theorie.

In der Realität sind genau solche Fehler in den Programmen, im Betriebssystem oder nicht zuletzt in der Benutzung der letzte Weg, um doch noch an die geheimen Informationen zu gelangen – auch deshalb sollte Sie dieses Kompendium bis hierhin gelesen haben.

In jedem Beispiel dieses Kompendiums haben Sie gesehen, dass zwischen dem geheimen und dem öffentlichen Schlüssel eine Verbindung besteht. Nur wenn beide zueinander passen, können geheime Botschaften entschlüsselt werden.

Das Geheimnis dieser mathematischen Verbindung müssen Sie nicht unbedingt kennen – Gpg4win funktioniert für Sie auch so. Man kann diese komplexe mathematische Methode aber auch als Normalsterblicher und Nichtmathematiker verstehen. Sie müssen eigentlich nur einfache Additionen (z.B. $2 + 3$) und Multiplikationen (z.B. $5 * 7$) beherrschen. Allerdings in einer ganzen anderen Rechenmethode als der, die Sie im Alltag benutzen. Es gehört sowohl zur Sicherheitsphilosophie der Kryptographie wie auch zum Prinzip der Freien Software, dass es keine geheim gehaltenen Methoden und Algorithmen gibt. Letztendlich versteht man auch erst dann wirklich, warum GnuPG (die eigentliche Maschinerie hinter Gpg4win) sicher ist.

Hier beginnt also sozusagen die Kür nach dem Pflichtteil:

25 GnuPG und das Geheimnis der großen Zahlen

Kryptographie für Nicht-Mathematiker

Es ist schon versucht worden, den RSA Algorithmus, auf dem GnuPG basiert[1], zu „knacken“, also einen privaten Schlüssel zu berechnen, wenn man lediglich den öffentlichen Schlüssel kennt. Diese Berechnung ist aber noch nie für Schlüssellängen (1024 Bit und mehr), die in GnuPG verwendet werden, gelungen. Es ist zwar theoretisch möglich, aber praktisch nicht durchführbar! Denn selbst bei genügend vorhandener Zeit (viele Jahre) und Abertausenden von vernetzten Rechnern würde niemals genügen Speicher zur Verfügung stehen, um den letzten Schritt dieser Berechnung durchführen zu können.

Es kann allerdings durchaus möglich sein, dass eines Tages eine geniale Idee die Mathematik revolutioniert und eine schnelle Lösung des mathematischen Problems, welches hinter RSA steckt, liefert. Dies wird aber wohl kaum von heute auf morgen geschehen. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) veröffentlicht von Zeit zu Zeit Prognosen und Einschätzungen, welche Schlüssellängen noch wieviele Jahre für absolute Geheimhaltung benutzt werden sollen. GnuPG überschreitet mit seinen Standardeinstellungen diese Mindestanforderungen. Wie im vorigen Kapitel schon angerissen, ist die Mathematik der mit Abstand sicherste Teil der praktisch angewandten Kryptographie.

[1] Es wird hier RSA als Beispiel verwendet, da RSA einfacher zu verstehen ist, als der Elgamal Algorithmus, der als Voreinstellung von GnuPG benutzt wird.

Im Folgenden erfahren Sie, wie diese mathematische Methode funktioniert. Nicht in allen Einzelheiten (das würde den Rahmen dieser Anleitung bei weitem sprengen), aber doch so, dass Sie bei etwas Mitrechnen selbst mathematisch korrekt ver- und entschlüsseln können und dabei das „Geheimnis der großen Zahlen“ entdecken.

Man kann diese komplexe mathematische Methode auch als Normalsterblicher und Nichtmathematiker verstehen. Sie müssen nur einfache Additionen und Multiplikationen beherrschen. Wie gesagt: Hier beginnt der Kürteil, und bei der Kür geht es immer etwas mehr zur Sache als im Pflichtprogramm. Letztendlich versteht man dann aber, warum GnuPG sicher ist.

Eine Begriffsklärung vorneweg:

> Ein **Algorithmus** ist eine mathematische Prozedur zur Veränderung oder Transformation von Daten oder Informationen.
>
> **Arithmetik** ist die Methode, nach der Sie Zahlen addieren und multiplizieren.

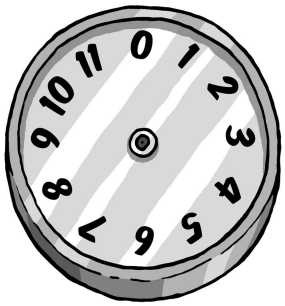
Die Verschlüsselung mit GnuPG basiert auf dem sogenannten RSA-Algorithmus[2]. RSA steht für die Nachnamen von Ron Rivest, Ami Shamir und Ben Adleman, die diesen Algorithmus im Jahr 1978 entdeckt haben. Dieser Algorithmus verwendet einen Typ der Arithmetik, die Rechnen mit Restklassen oder „Modulo-Arithmetik“ heißt.

[2] RSA ist eigentlich optional, da aus Patentgründen der Elgamal Algorithmus, beruhend auf dem schwieriger zu erklärenden Problem des diskreten Logarithmus, als Standard in GnuPG verwendet wird.

25.1 Das Rechnen mit Restklassen

Wenn man mit Restklassen rechnet, so bedeutet dies, dass man nur mit dem „Rest“ rechnet, der nach einer ganzzahligen Teilung durch eine bestimmte Zahl übrigbleibt. Diese Zahl, durch die geteilt wird, nennt man den „Modul“ oder die „Modulzahl“. Wenn Sie beispielsweise mit dem Teiler oder der Modulzahl 5 rechnen, sagen man auch, „ich rechne modulo 5“.

Wie das Rechnen mit Restklassen – auch Modulo-Arithmetik oder Kongruenzrechnung genannt – funktioniert, kann man sich gut klarmachen, wenn man sich das Zifferblattes einer Uhr vorstellt:



Diese Uhr ist ein Beispiel für das Rechnen mit modulo 12 (der Teiler ist also 12) – eine Uhr mit einem normalen Zifferblatt, allerdings mit einer 0 anstelle der 12. Sie können damit Modulo-Arithmetik betreiben, indem Sie einfach den gedachten Zeiger bewegen.

Um beispielsweise $3+2$ zu rechnen, beginnen Sie bei der Ziffer 2 und drehen den Zeiger um 3 Striche weiter (oder Sie starten bei der 3 und drehen 2 Striche weiter, was natürlich auf dasselbe hinausläuft) Das Ergebnis ist 5.

Zählt man auf diese Weise $7 + 8$ zusammen, erhält man 3. Denn 3 ist der Rest, wenn man 15 (also $7 + 8$) durch 12 teilt. Um 5 mit 7 zu multiplizieren, beginnt man bei 0 und dreht 7 mal jeweils um 5 Striche weiter (oder auch bei 0 beginnend 5 mal um 7 Striche). In beiden Fällen bleibt der Zeiger bei 11 stehen. Denn 11 ist der Rest, wenn 35 (also $7 * 5$) durch 12 geteilt wird.

Beim Rechnen mit Restklassen addieren und teilen Sie Zahlen also nach den normalen Regeln der Alltagsarithmetik, verwenden dabei jedoch immer nur den Rest nach der Teilung. Um anzuzeigen, dass Sie nach den Regeln der Modulo-Arithmetik und nicht nach denen der üblichen Arithmetik rechnen, schreibt man den Modul (Sie wissen schon – den Teiler) dazu. Man sagt dann z.B. „4 modulo 5“, schreibt aber kurz „$4 \bmod 5$“.

Bei Modulo-5 z.B. hat man dann eine Uhr, auf deren Zifferblatt es nur die 0, 1, 2, 3 und 4 gibt. Also:

$$4 \bmod 5 + 3 \bmod 5 = 7 \bmod 5 = 2 \bmod 5$$

Anders ausgedrückt, ist in der Modulo-5 Arithmetik das Ergebnis aus 4 plus 3 gleich 2. Sie können also auch schreiben:

$$9 \bmod 5 + 7 \bmod 5 = 16 \bmod 5 = 1 \bmod 5$$

Sie sehen auch, dass es egal ist, in welcher Reihenfolge Sie vorgehen, weil Sie nämlich auch schreiben können:

$$9 \bmod 5 + 7 \bmod 5 = 4 \bmod 5 + 2 \bmod 5 = 6 \bmod 5 = 1 \bmod 5$$

Denn 4 ist dasselbe wie 9, und 2 dasselbe wie 7, da Sie sich ja nur für den jeweiligen Rest nach der Teilung durch 5 interessieren. Daran wird deutlich, dass Sie bei dieser Art der Arithmetik jederzeit 5 oder ein Vielfaches von 5, wie 10, 15 und so weiter nehmen können, und das Ergebnis stets dasselbe ist.

Das funktioniert auch beim Multiplizieren (Malnehmen).

Ein Beispiel:

$$4 \bmod 5 * 2 \bmod 5 = 8 \bmod 5 = 3 \bmod 5$$

Ebenso können Sie schreiben:

$$9 \bmod 5 * 7 \bmod 5 = 63 \bmod 5 = 3 \bmod 5$$

da Sie einfach 60, also $5 * 12$, abziehen können.

Man könnte aber auch schreiben:

$$9 \bmod 5 * 7 \bmod 5 = 4 \bmod 5 * 2 \bmod 5 = 8 \bmod 5 = 3 \bmod 5$$

denn 4 entspricht 9, und 2 entspricht 7, wenn Sie nur den Rest nach Teilung durch 5 betrachten.

Wiederum können Sie feststellen, dass es egal ist, wenn Sie das Vielfache von 5 einfach weglassen.

Da dadurch alles einfacher wird, machen Sie das, bevor Sie Zahlen addieren oder multiplizieren. Das bedeutet, dass Sie sich lediglich um die Zahlen 0, 1, 2, 3 und 4 kümmern müssen, wenn Sie mit der Modulo-5 Arithmetik rechnen. Denn Sie können ja alles, was durch 5 teilbar ist, weglassen. Dazu noch drei Beispiele:

$$5 \bmod 11 * 3 \bmod 11 = 15 \bmod 11 = 4 \bmod 11$$

$$2 \bmod 7 * 4 \bmod 7 = 1 \bmod 7$$

$$13 \bmod 17 * 11 \bmod 17 = 7 \bmod 17$$

Das letzte Beispiel wird klar, wenn man bedenkt, dass in normaler Arithmetik gerechnet $13 * 11 = 143$ und $143 = 8 * 17 + 7$ ist.

25.2 RSA-Algorithmus und Rechnen mit Restklassen

Computer speichern Buchstaben als Zahlen. Alle Buchstaben und Symbole auf der Computertastatur werden in Wirklichkeit als Zahlen gespeichert, die zwischen 0 und 255 liegen.

Sie können also eine Nachricht auch in eine Zahlenfolge umwandeln. Nach welcher Methode (oder Algorithmus) dies geschieht, wird im nächsten Abschnitt beschrieben. Darin wird Ihnen die Methode vorgestellt, nach der die Verschlüsselung mit GnuPG funktioniert: den RSA Algorithmus. Dieser Algorithmus wandelt eine Zahlenfolge (die ja eine Nachricht darstellen kann) so in eine andere Zahlenfolge um (Transformation), dass die Nachricht dabei verschlüsselt wird. Wenn man dabei nach dem richtigen Verfahren vorgeht, wird die Nachricht sicher kodiert und kann nur noch vom rechtmäßigen Empfänger dekodiert werden. Das sind die Grundlagen des RSA Algorithmus:

Sie selbst haben bei der Installation von Gpg4win während der Eingabe Ihrer Passphrase zwei große Primzahlen erzeugt, ohne es zu bemerken (dieser werden mit p und q bezeichnet). Nur Sie – oder in der Praxis Ihr Rechner – kennen diese beiden Primzahlen, und Sie müssen für ihre Geheimhaltung sorgen.

Es werden daraus nun drei weitere Zahlen erzeugt:

Die erste Zahl ist das Ergebnis der Multiplikation der beiden Primzahlen, also ihr Produkt. Dieses Produkt wird als Modulus und dem Buchstaben n bezeichnet. Dies ist der Modul mit dem Sie später immer rechnen werden.

Die zweite Zahl ist der sogenannte öffentliche Exponent und eine Zahl an die bestimmte Anforderungen gestellt werden (teilerfremd zu $(p-1)(q-1)$); sie wird mit e bezeichnet. Häufig wird hier 3, 41 oder 65537 benutzt.

Die dritte Zahl wird errechnet aus dem öffentlichem Exponent (der zweiten Zahl) und den beiden Primzahlen. Diese Zahl ist der geheime Exponent und wird mit d bezeichnet. Die komplizierte Formel zur Berechnung lautet:

$$d = e^{-1} \bmod (p-1)(q-1)$$

Die erste und die zweite Zahl werden veröffentlicht – das ist Ihr öffentlicher Schlüssel. Beide werden dazu benutzt, Nachrichten zu verschlüsseln. Die dritte Zahl muss von Ihnen geheimgehalten werden – es ist Ihr geheimer Schlüssel. Die beiden Primzahlen werden danach nicht mehr benötigt.

Wenn eine verschlüsselte Nachricht empfangen wird, kann sie entschlüsselt werden mit Hilfe der ersten (n) und der dritten Zahl (d). Nur der Empfänger kennt beide Schlüsselteile – seinen öffentlichen und seinen geheimen Schlüssel. Der Rest der Welt kennt nur den öffentlichen Schlüssel (n und e).

Die Trick des RSA Algorithmus liegt nun darin, dass es unmöglich ist, aus dem öffentlichen Schlüsselteil (n und e) den geheimen Schlüsselteil (d) zu errechnen und damit die Botschaft zu entschlüsseln – denn: Nur wer im Besitz von d ist, kann die Botschaft entschlüsseln.

25.3 RSA Verschlüsselung mit kleinen Zahlen

Sie verwenden hier erst einmal kleine Zahlen, um deutlich zu machen, wie die Methode funktioniert. In der Praxis verwendet man jedoch viel größere Primzahlen, die aus zig Ziffern bestehen.

Nehmen Sie die Primzahlen 7 und 11. Damit verschlüsseln Sie Zahlen – oder Buchstaben, was für den Rechner dasselbe ist – nach dem RSA Algorithmus.

Und zwar erzeugen Sie zunächst den öffentlichen Schlüssel.

Die erste Zahl ist 77, nämlich das Ergebnis der Multiplikation der beiden Primzahlen, 7 und 11. 77 dient Ihnen im weiteren Verlauf als Modulus zur Ver- und Entschlüsselung.

Die zweite Zahl ist der öffentliche Exponent. Sie wählen hier 13.

Die dritte Zahl ist der geheime Schlüssel. Diese Zahl wird in einem komplizierten Verfahren errechnet, welches Sie jetzt erklärt bekommen:

Zunächst ziehen Sie von Ihren Primzahlen 7 und 11 jeweils die Zahl 1 ab (also $7-1$ und $11-1$) und multiplizieren die beiden resultierenden Zahlen miteinander. In dem Beispiel ergibt das 60: $(7-1)*(11-1) = 60$. 60 ist Ihre Modulzahl für die weiterführende Berechnung des geheimen Schlüssels (sie ist aber nicht mit dem eigentlichen Modulus 77 zu verwechseln).

Sie suchen jetzt eine Zahl, die multipliziert mit dem öffentlichen Schlüssel die Zahl 1 ergibt, wenn man mit dem Modul 60 rechnet:

$$13 \bmod 60 * \ ? \ \bmod 60 = 1 \bmod 60$$

Die einzige Zahl, die diese Bedingung erfüllt, ist 37, denn

$$13 \bmod 60 * 37 \bmod 60 = 481 \bmod 60 = 1 \bmod 60$$

37 ist die einzige Zahl, die multipliziert mit 13 die Zahl 1 ergibt, wenn man mit dem Modul 60 rechnet.

Sie verschlüsseln mit dem öffentlichen Schlüssel eine Nachricht

Nun zerlegen Sie die Nachricht in eine Folge von Zahlen zwischen 0 und 76, also 77 Zahlen, denn sowohl Verschlüsselung als auch Entschlüsselung verwenden den Modul 77 (das Produkt aus den Primzahlen 7 und 11).

Jede einzelne dieser Zahlen wird nun nach der Modulo-77 Arithmetik 13 mal mit sich selbst multipliziert. Sie erinnern sich: Die 13 ist ja Ihr öffentlicher Schlüssel.

Nehmen Sie ein Beispiel mit der Zahl 2: Sie wird in die Zahl 30 umgewandelt, weil $2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 8192 = 30 \mod 77$ sind.

Ein weiteres Beispiel: 75 wird in die Zahl 47 umgewandelt, denn 75 wird 13 mal mit sich selbst multipliziert und durch 77 geteilt, so dass der Rest 47 entsteht.

Wenn man eine solche Rechnung für alle Zahlen zwischen 0 und 76 durchführt und die Ergebnisse in eine Tabelle einsetzt, sieht diese so aus:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	30	38	53	26	62	35	50	58
10	10	11	12	41	49	64	37	73	46	61
20	69	21	22	23	52	60	75	48	7	57
30	72	3	32	33	34	63	71	9	59	18
40	68	6	14	43	44	45	74	5	20	70
50	29	2	17	25	54	55	56	8	16	31
60	4	40	13	28	36	65	66	67	19	27
70	42	15	51	24	39	47	76			

Tabelle 25.1:

In der linken Spalte stehen die 10er-Stellen, in der oberen Zeile die 1er-Stellen.

Entschlüsseln Sie eine Nachricht mit dem privaten Schlüssel

Um das Beispiel mit der 2 von oben umzukehren, also die Nachricht zu dekodieren, multiplizieren Sie 30 (die umgewandelte 2) 37 mal mit sich selbst. Das Ergebnis wird modulo der Modulzahl 77 gerechnet. Sie erinnern sich: 37 ist der geheime Schlüssel.

Diese wiederholte Multiplikation ergibt eine Zahl die $2 \bmod 77$ ist. Das andere Beispiel: Die Zahl $47 \bmod 77$ wird zur Zahl $75 \bmod 77$ dekodiert.

Tabelle 25.2 zeigt die genaue Zuordnung der 77 Zahlen zwischen 0 und 76.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	51	31	60	47	41	28	57	37
10	10	11	12	62	42	71	58	52	39	68
20	48	21	22	23	73	53	5	69	63	50
30	2	59	32	33	34	7	64	16	3	74
40	61	13	70	43	44	45	18	75	27	14
50	8	72	24	4	54	55	56	29	9	38
60	25	19	6	35	15	65	66	67	40	20
70	49	36	30	17	46	26	76			

Tabelle 25.2: Zahlentransformation modulo77, unter Verwendung des geheimen Schlüssels 37

Um eine Zahl mit Tabelle 25.2 zu transformieren, gehen Sie nach der gleichen Methode vor wie bei Tabelle 25.1. Ein Beispiel: 60 wird transformiert in die Zahl in Zeile 60 und Spalte 0. Also wird 60 zu 25 transformiert.

Das überrascht nicht, denn wenn man davon ausgeht, dass Sie bei der Umwandlung von 25 mit Hilfe von Tabelle 25.1 als Ergebnis 60 erhalten, dann sollten Sie auch bei der Transformation von 60 mit Hilfe von Tabelle 25.2 zum Ergebnis 25 gelangen. Dabei haben Sie den öffentlichen Schlüssel, 13, zur Umwandlung bzw. Kodierung einer Zahl verwendet, und den geheimen Schlüssel 37, um sie zurückzuwandeln bzw. zu dekodieren. Sowohl für die Verschlüsselung als auch für die Entschlüsselung haben Sie sich der Modulo-77 Arithmetik bedient.

Zusammenfassung

Sie haben…

- durch den Rechner zwei zufällige Primzahlen erzeugen lassen;
- daraus das Produkt und den öffentlichen und den geheimen Subkey gebildet;
- gezeigt, wie man mit dem öffentlichen Schlüssel Nachrichten verschlüsselt;
- gezeigt, wie man mit dem geheimen Schlüssel Nachrichten entschlüsselt.

Diese beiden Primzahlen können so groß gewählt werden, dass es unmöglich ist, sie einzig aus dem öffentlich bekannt gemachten Produkt zu ermitteln. Das begründet die Sicherheit des RSA Algorithmus.

Sie haben gesehen, dass die Rechnerei sogar in diesem einfachen Beispiel recht kompliziert geworden ist. In diesem Fall hat die Person, die den Schlüssel öffentlich gemacht hat, die Zahlen 77 und 13 als öffentlichen Schlüssel bekanntgegeben. Damit kann jedermann dieser Person mit der oben beschriebenen Methode – wie im Beispiel der Tabelle 25.1 – eine verschlüsselte Zahl oder Zahlenfolge schicken. Der rechtmäßige Empfänger der verschlüsselten Zahlenfolge kann diese dann mit Hilfe der Zahl 77 und dem geheimen Schlüssel 37 dekodieren.

In diesem einfachen Beispiel ist die Verschlüsselung natürlich nicht sonderlich sicher. Es ist klar, dass 77 das Produkt aus 7 und 11 ist.

Folglich kann man den Code in diesem einfachen Beispiel leicht knacken. Der scharfsinnige Leser wird auch bemerkt haben, dass etliche Zahlen, z.B. die Zahl 11 und ihr Vielfaches (also 22, 33 etc.) und die benachbarten Zahlen sich in sich selbst umwandeln.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	51	31	60	47	41	28	57	37
10	10	11	12	62	42	71	58	52	39	68
20	48	21	22	23	73	53	5	69	63	50
30	2	59	32	33	34	7	64	16	3	74
40	61	13	70	43	44	45	18	75	27	14
50	8	72	24	4	54	55	56	29	9	38
60	25	19	6	35	15	65	66	67	40	20
70	49	36	30	17	46	26	76			

Tabelle 25.3:

Das erscheint als ein weiterer Schwachpunkt dieser Verschlüsselungsmethode: Man könnte annehmen, dass die Sicherheit des Algorithmus dadurch beeinträchtigt würde. Doch stellen Sie sich nun vor, das Produkt zweier grosser Primzahlen, die auf absolut willkürliche Art und Weise gewählt werden, ergäbe

```
114,381,625,757,888,867,669,235,779,976,146,612,010,
218,296,721,242,362,562,561,842,935,706,935,245,733,
897,830,597,123,563,958,705,058,989,075,147,599,290,
026,879,543,541
```

Hier ist überhaupt nicht mehr ersichtlich, welche die beiden zugrunde liegenden Primzahlen sind. Folglich ist es sehr schwierig, aufgrund des öffentlichen Schlüssels den geheimen Schlüssel zu ermitteln. Selbst den schnellsten Rechner der Welt würde es gewaltige Probleme bereiten, die beiden Primzahlen zu errechnen.

Man muss die Primzahlen also nur groß genug wählen, damit ihre Berechnung aus dem Produkt so lange dauert, dass alle bekannten Methoden daran in der Praxis scheitern. Außerdem nimmt der Anteil der Zahlen, die in sich selbst transformiert werden – wie man sie oben in den Tabellen 25.1 und 25.2 findet – stetig ab, je größer die Primzahlen werden. Von Primzahlen in der Grössenordnung, die Sie in der Praxis bei der Verschlüsselung verwenden, ist dieser Teil ist so klein, dass der RSA Algorithmus davon in keiner Weise beeinträchtigt wird.

Je größer die Primzahlen, desto sicherer die Verschlüsselung. Trotzdem kann ein normaler PC ohne weiteres das Produkt aus den beiden großem Primzahlen bilden. Kein Rechner der Welt dagegen kann aus diesem Produkt wieder die ursprünglichen Primzahlen herausrechnen – jedenfalls nicht in vertretbarer Zeit.

25.4 Die Darstellung mit verschiedenen Basiszahlen

Um zu verstehen, wie Nachrichten verschlüsselt werden, sollte man wissen, wie ein Rechner Zahlen speichert und vor allem, wie sie in unterschiedlichen Zahlenbasen dargestellt werden können.

Dazu machen Sie sich zunächst mit den Zahlenpotenzen vertraut.

Zwei hoch eins, das man als 2^1 darstellt, ist gleich 2; zwei hoch drei, dargestellt als 2^3, ist $2*2*2 = 8$; zwei hoch zehn, dargestellt als 2^{10}, ist $2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 1024$.

Jede Zahl hoch 0 ist gleich 1, z.B. $2^0 = 1$ und $5^0 = 1$. Verallgemeinert bedeutet dies, dass eine potenzierte Zahl so oft mit sich selbst multipliziert wird, wie es die Hochzahl (Potenz) angibt.

Das Konzept einer Zahlenbasis veranschaulicht z.B. ein Kilometerzähler im Auto: Das rechte Rad zählt nach jedem Kilometer eine Stelle weiter und zwar nach der vertrauten Abfolge der Zahlen

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2,...

und so weiter. Jedesmal, wenn das rechte Rad wieder 0 erreicht, zählt das Rad links davon eine Stelle hoch. Und jedesmal, wenn dieses zweite Rad die 0 erreicht, erhöht das Rad links davon um eins . . . und so weiter.

Das rechte Rad zählt die einzelnen Kilometer. Wenn es eine 8 angezeigt, dann sind dies 8 Kilometer. Das Rad links davon zeigt jeweils die vollen zehn Kilometer an: eine 5 bedeutet 50 Kilometer. Dann folgen die Hunderter: Steht dort 7, dann bedeutet dies 700 Kilometer.

Nach dem gleichen Prinzip stellen Sie ja auch Ihre normale Zahlen mit den Ziffern 0 bis 9 dar.

„578“, z.B., bedeutet $5 * 10^2 + 7 * 10^1 + 8 * 10^0 = 500 + 70 + 8$, und dies entspricht 578.

Hier haben Sie die „5“ stellvertretend für fünfhundert, „7“ für siebzig und „8“ für acht. In diesem Fall ist die Basis 10, eine für Sie vertraute Basis.

Also steht die rechte Ziffer für die Einer der betreffenden Zahl (d.h. sie wird mit 1 multipliziert), die Ziffer links davon steht für die Zehner (d.h. wird mit 10 multipliziert), die nächste Ziffer wiederum für die Hunderter (d.h. sie wird mit 100 multipliziert) und so weiter. Da man Zahlen normalerweise zur Basis 10 darstellen, machen Sie sich nicht die Mühe, die Basis extra anzugeben. Formal würde man dies bei der Zahl 55 mit der Schreibweise 55_{10} anzeigen, wobei die tiefgestellte Zahl die Basis anzeigt.

Wenn Sie Zahlen nicht zur Basis 10 darstellen, so müssen Sie dies mit Hilfe einer solchen tiefgestellten Basiszahl anzeigen.

Angenommen, die Anzeige des Kilometerzählers hätte statt der Ziffern 0 bis 9 nur noch 0 bis 7. Das rechte Rädchen würde nach jedem Kilometer um eine Ziffer höher zählen, wobei die Zahlenfolge so aussehen würde:

$$0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, ...$$

Ihr Tacho zur Basis 8 stellt z.B. folgende Zahl dar:

$$356$$

Die 6 auf dem rechten Rädchen zählt einzelne Kilometer, also $6 * 8^0 = 6$ Kilometer.
Die 5 auf dem Rädchen daneben für $5 * 8^1$, also 40 Kilometer.
Die 3 links steht für je 64 Kilometer pro Umdrehung, also hier $3 * 8^2$ Kilometer.

So rechnet man also mit Zahlen zur Basis 8. Ein Beispiel: 728 bedeutet $7 * 8^1 + 2 * 8^0$, und das ist gleich „58“. Bei dieser Art der Darstellung steht die „2“ aus der 72 für 2, aber die „7“ steht für $7 * 8$.

Größere Zahlen werden schrittweise genauso aufgebaut, so dass 453_8 eigentlich $4 * 64 + 5 * 8 + 3$ bedeutet, was 299_{10} ergibt.

Bei 453_8 steht die „3“ für 3, die „5“ für $5 * 8$ und die „4“ für $4 * 64$, wobei sich die „64“ wiederum aus $8 * 8$ herleitet.

Im angeführten Beispiel werden die Ziffern, von rechts nach links gehend, mit aufsteigenden Potenzen von 8 multipliziert. Die rechte Ziffer wird mit 8 hoch 0 (das ist 1) multipliziert, die links daneben mit 8 hoch 1 (das ist 8), die nächste links davon mit 8 hoch 2 (das ist 64) und so weiter.
Wenn man Zahlen zur Basis 10 darstellt, gibt es keine höhere Ziffer als 9 (also 10 minus 1). Sie verfügen also über keine Ziffer, die 10 oder eine größere Zahl darstellt. Um 10 darzustellen, brauchen Sie zwei Ziffern, mit denen Sie dann die „10“ schreiben können.
Sie haben also nur die Ziffern 0 bis 9.

So ähnlich ist es, wenn Sie mit der Basiszahl 8 rechnen: Dann haben Sie nur die Ziffern 0 bis 7. Wollen Sie zu dieser Basis eine höhere Zahl als sieben darstellen, müssen Sie wieder zwei Ziffern verwenden. Z.B. „9“ schreibt man als 11_8, „73“ schreibt man als 111_8.

Rechner speichern Zahlen als eine Folge von Nullen und Einsen. Man nennt dies Binärsystem oder Rechnen mit der Basiszahl 2, weil Sie nur die Ziffern 0 und 1 verwenden. Stellen Sie sich vor, Sie würden die Kilometer mit einem Tachometer zählen, auf dessen Rädchen sich nur zwei Ziffern befinden: 0 und 1. Die Zahl 10101_2 z.B. bedeutet im Binärsystem

$$1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 1 * 16 + 0 * 8 + 1 * 4 + 0 * 2 + 1 = 21$$

.

In der Computerei verwendet man auch Gruppen von acht Binärziffern, das wohlbekannte Byte. Ein Byte kann Werte zwischen 0 - dargestellt als Byte 00000000_2 – und 255 – dargestellt als Byte 11111111_2 – annehmen. Ein Byte stellt also Zahlen zur Basis $2^8 = 256$ dar.

Zwei weitere Beispiele:

$$10101010_2 = 170$$

und

$$00000101_2 = 5$$

.

Da der Rechner die Buchstaben, Ziffern und Satzzeichen als Bytes speichert, schauen Sie sich an, welche Rolle dabei die Darstellung zur Basis 256 spielt.

Nehmen Sie die Silbe „un“. Das „u“ wird im Rechner als 117 gespeichert und das „n“ als 110.

Diese Zahlenwerte sind für alle Rechner standardisiert und werden ASCII-Code genannt. Um alle Zahlen und Symbole darstellen zu können, benötigen Sie auf dem Rechner die 256 Zahlen von 0 bis 255.

Sie können also die Silbe „un“ durch die Zahl $117 * 256 + 110$ darstellen.
Entsprechend würde man die Buchstabenfolge „und“ mit der Zahl $117 * 65536 + 110 * 256 + 100$ darstellen, denn das „d“ wird durch 100 repräsentiert.
Sie haben hier also Zahlen und Symbole, die auf der Computertastatur als normale Zahlen zur Basis 10 stehen, intern durch Zahlen zur Basis 256 repräsentiert.

Entsprechend können Sie aus jeder Nachricht eine große Zahl machen. Aus einer langen Nachricht wird also eine gewaltig große Zahl. Und diese sehr große Zahl wollen Sie nun nach dem RSA Algorithmus verschlüsseln.

Sie dürfen allerdings dabei die Zahl, zu der die Nachricht verschlüsselt wird, nicht größer werden lassen als das Produkt der Primzahlen (Modulus). Ansonsten bekommen Sie Probleme, wie Sie gleich noch sehen werden.

Die folgende Prozedur umfasst mehrere Schritte, die hier zunächst zusammengefasst werden und anschließend in Einzelschritten dargestellt werden:

1. Die Nachricht *aba, cad, ada* wandeln Sie – wie gesehen – in Zahlen um.
2. Diese Darstellung, beispielhaft zur Basis $2^2 = 4$ (statt $2^8 = 256$), wandeln Sie in eine Darstellung zur Basis 10 um, damit Sie zur Verschlüsselung die Tabelle 25.1 benutzen können, in denen die Zahlen ja auch auf 10er-Basis dargestellt werden. Dabei entsteht eine kodierte Nachricht zur Basis 10.
3. Um die Kodierung im Vergleich zum „Klartext“ zu erkennen, rechnen Sie die zur Basis 10 kodierte Nachricht auf die Basis 4 zurück und wandeln sie dann wieder in eine Buchstabensequenz.
4. So entsteht aus der Nachricht *aba, cad, ada* die verschlüsselte Nachricht *dbb, ddd, dac*.

Und nun ausführlich:

1. Die Nachricht *aba, cad, ada* wandeln Sie – wie gesehen – in Zahlen um.

   Angenommen, Sie beschränken sich bei den Nachrichten auf die 4 Buchstaben a, b, c und d. In diesem – wirklich sehr einfachen – Beispiel können Sie die vier Buchstaben durch die Zahlenwerte 0, 1, 2 und 3 darstellen, und haben dann

   $$a = 0, b = 1, c = 2 \text{ und } d = 3$$

   .

   Verschlüsseln Sie nun die Nachricht „abacadaca“. Sie kodieren diese Nachricht mit Hilfe der Primzahlen 7 und 11, mit dem öffentlichen Schlüssel 77 und 13 und dem dazugehörenden geheimen Schlüssel 37. Dieses Beispiel kennen Sie bereits aus dem früheren Kapitel: Sie haben damit die Tabellen 25.1 und 25.2 konstruiert.

2. Diese Darstellung zur Basis 4 wandeln Sie in eine Darstellung zur Basis 10 um. Damit können Sie zur Verschlüsselung die Tabelle 25.1 benutzen, in denen die Zahlen ja auch auf 10er-Basis dargestellt werden.

   Weil Sie vier Buchstaben für die Nachricht verwenden, rechnen Sie zur Basis 4. Für die Rechnung modulo 77 müssen Sie die Nachricht in Stücke von je drei Zeichen Länge zerlegen, weil die größte dreiziffrige Zahl zur Basis 4 die 333_4 ist. Zur Basis 10 hat diese Zahl den Wert 63.

   Würden Sie stattdessen die Nachricht in vier Zeichen lange Stücke zerlegen, würde 3333_4 den Wert 76_{10} übersteigen und es würden unerwünschte Doppeldeutigkeiten entstehen.
   Folglich würde die Nachricht in dreiziffrigen Stücken nun

   $$aba, cad, aca$$

   ergeben. Geben Sie den Zeichen nun ihre Zahlenwerte und vergessen dabei nicht, dass die Stücke dreiziffrige Zahlen zur Basis 4 darstellen.

   Da Sie die Buchstaben durch die Zahlen a = 0, b = 1, c = 2, d = 3 darstellen, wird die Nachricht zu:

   $$010_4, 203_4, 020_4$$

   Zur Basis 10 wird diese Nachricht durch die Zahlenfolge 4, 35, 8 dargestellt. Warum? Nehmen Sie z.B. das mittlere Stück 203_4:

   $$\begin{array}{lll} 3 * 4^0, & \text{also } 3 * 1, & \text{also } 3. \\ 0 * 4^1, & \text{also } 0 * 4, & \text{also } 0. \\ 2 * 4^2, & \text{also } 2 * 16, & \text{also } 32. \end{array}$$

3. Jetzt können Sie zur Verschlüsselung die Tabelle 25.1 von Seite 146 benutzen, die ja zur Basis 10 berechnet wurde. Diese Tabelle benutzen wir, weil Sie mit dem schon bekannten Schlüsselpaar arbeiten wollen. Dabei entsteht eine kodierte Nachricht zur Basis 10.

   Zum Verschlüsseln der Nachricht nehmen Sie jetzt die o.g. Tabelle 25.1 zur Hilfe. Die Nachricht wird nun zu der Zahlenfolge 53, 63, 50 (zur Basis 10).

4. Wiederum zur Basis 4 konvertiert, entsteht die verschlüsselte Nachricht.

   Wird sie nun wieder zur Basis 4 konvertiert, ergibt die Nachricht nun $311_4, 333_4, 302_4$. Konvertiert man diese zu einer Buchstabensequenz, erhält man dbb, ddd, dac, was sich nun erheblich von der ursprünglichen Nachricht unterscheidet.

   Man kehrt nun also den Prozess um und transformiert die Zahlenfolge 53, 63, 50 mit Tabelle 25.2 und erhält die Sequenz 4, 35, 8. Und das entspricht, als Zahlenfolge genau der ursprünglichen Nachricht.

   Anhand der Tabellen 25.1 und 25.2 können Sie ebensogut Nachrichten unter Verwendung des geheimen Schlüssels (d.h. erst Tabelle 25.2 benutzen) verschlüsseln, dann mit dem öffentlichen Schlüssel (d.h. Tabelle 25.1 als zweites benutzen) dekodieren und damit Ihre ursprüngliche Zahl wieder herstellen. Das bedeutet, dass der Inhaber des geheimen Schlüssels damit Nachrichten unter Verwendung des RSA Algorithmus verschlüsseln kann. Damit ist bewiesen, dass sie eindeutig nur von ihm stammen können.

Fazit:

Wie Sie gesehen haben, ist die ganze Angelegenheit zwar im Detail kompliziert, im Prinzip aber durchaus nachvollziehbar. Sie sollen schließlich nicht einer Methode einfach nur vertrauen, sondern – zumindest ansatzweise – ihre Funktionsweise durchschauen. Sehr viele tiefergehende Details sind leicht in anderen Büchern (z.B.: R. Wobst, „Abenteuer Kryptologie“) oder im Internet zu finden.

Immerhin wissen Sie nun: Wenn jemand sich an Ihren verschlüsselten E-Mails zu schaffen macht, ist er durchaus so lange damit beschäftigt, dass er dann keine Lust mehr haben dürfte diese dann noch zu lesen. . .

Teil III

Anhang

A Glossar

[Dieses Glossar ist noch in Arbeit.]

OpenPGP ...

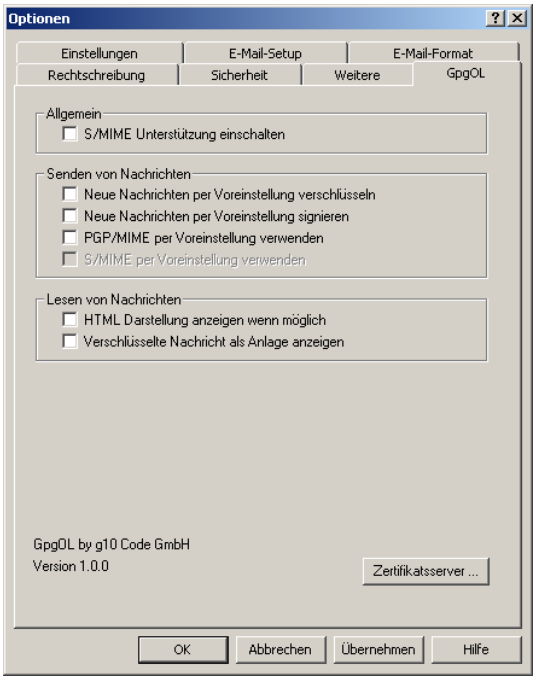
X.509 ...

B Hinweise zur Outlook-Programmerweiterung GpgOL

GpgOL ist eine Programmerweiterung für Microsoft Outlook, es integriert dort die Bedienung von GnuPG.

Da Outlook ein proprietäres Produkt, also nicht als Freie Software mit Quelltext verfügbar ist, hat die Integration eine Reihe von „Ecken und Kanten”. Oder mit anderen Worten: Die Bedienung ist nicht so komfortabel wie es beispielsweise E-Mail-Programme mit integrierter Verschlüsselungs- und Signaturkomponente bieten (z.B. KMail / Kontact).

GpgOL wird durch den Gpg4win-Installationsassistenten installiert. Beim nächsten Start von Outlook findet sich im Menü *Extras→Optionen* eine Karteikarte *GpgOL*:



Die Karteikarte *GpgOL* unterteilt sich in drei Bereiche:

S/MIME

1. **Allgemein:** Nach der Installation von Gpg4win ist die S/MIME Funktionalität in GpgOL deaktiviert. Damit ist die S/MIME Unterstützung von GnuPG gemeint. Outlook selbst unterstützt ebenfalls X.509 und S/MIME, arbeitet aber natürlich nicht mit den Gpg4win-Komponenten. Konkret heißt das, dass alle Einstellungen, das Zertifikatsmanagement und die Benutzerdialoge unterschiedlich sind. Es ist zu beachten, dass Outlook keine OpenPGP Unterstützung anbietet.

   Wichtig: Wenn Sie S/MIME in Outlook mit Gpg4win nutzen möchten, müssen Sie zuvor die GpgOL-Option *S/MIME Unterstützung einschalten* aktivieren.

2. **Senden von Nachrichten:**

   Die beiden ersten Optionen in diesem Bereich steuern, ob per Voreinstellung neue Nachrichten verschlüsselt und / oder signiert werden sollen. Sie können dies aber immer noch bei der Erstellung einer Nachricht individuell verändern. Lediglich die Schaltflächen sind schon entsprechend aktiviert.

   Die beiden letzten Optionen definieren, ob PGP/MIME *oder* S/MIME per Voreinstellung verwendet werden soll. Auch hier können Sie diese Entscheidung immer noch vor dem Senden jeder Nachricht nachträglich ändern.

3. **Lesen von Nachrichten:**

   Auch im Vorschaufenster entschlüsseln

   Soll im Vorschaufenster die entschlüsselte Fassung erscheinen, so ist diese Option einzuschalten. Sie sollten dabei bedenken, dass dadurch bereits beim Durchblättern durch Ihre Nachrichten die Entschlüsselungs- und Prüfroutinen ausgeführt werden. Das heißt, es werden Dialog zum Status der E-Mails angezeigt und ggf. werden Sie nach einer Passphrase zur Entschlüsselung gefragt.

   HTML-Darstellung anzeigen wenn möglich

   Diese Option kann benutzt werden, um die HTML-Version einer Nachricht anzuzeigen. Im Normalfall oder falls kein HTML-Format vorhanden ist, so wird die Nachricht im Text-Format dargestellt.

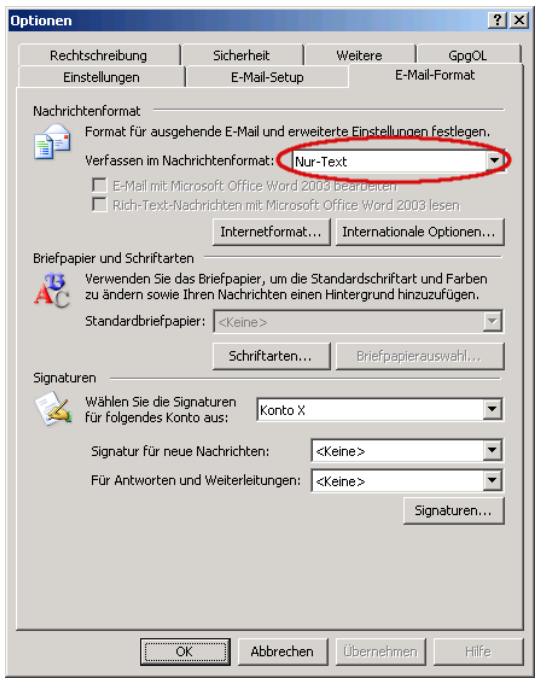
   Verschlüsselte Nachricht als Anlage anzeigen

   Der verschlüsselte Teil der Nachricht wird zusätzlich als Anhang angezeigt. Der Anwender kann so den verschlüsselten Teil separat speichern oder auf andere Weise weiterverarbeiten.

Alle Optionen sind nach einer Neuinstallation bereits sinnvoll vorbelegt.

Um verschlüsselte Nachrichten mit Outlook versenden zu können, müssen Sie sicherstellen, dass Sie **nicht** Microsoft Word zum Verfassen der Nachrichten benutzen.

Desweiteren ist anzuraten auf HTML Nachrichten zu verzichten. Sie können dies im Menüpunkt *Extras→Optionen* auf der Karteikarte *E-Mail-Format* kontrollieren. Das Nachrichtenformat sollte auf *Nur-Text* eingestellt sein (siehe markierter Bereich). Sollten Sie dennoch HTML für signierte oder verschlüsselte E-Mails verwenden, können dabei beim Empfänger die Formatierungsinformationen verloren gehen.



Beachten Sie: Mit der Programmerweiterung GpgOL von Gpg4win wird Outlook 2003/2007 um die Möglichkeit erweitert, mit E-Mails nach dem OpenPGP-Standard umzugehen. Gpg4win befähigt Outlook 2007 unter WindowsXP und Outlook 2003 generell AES-verschlüsselte S/MIME E-Mails zu entschlüsseln und zu erzeugen, da Gpg4win mit GnuPG eine eigene, von Outlook und Windows unabhängige Kryptokomponente mitbringt. Nur Outlook 2007 unter Windows Vista beherrscht AES-verschlüsselte S/MIME E-Mails auch ohne Gpg4win.

C GnuPG mit anderen E-Mail-Programme nutzen

Das Gpg4win-Kompendium geht vor allem auf das E-Mail-Programm Outlook ein. GnuPG ist jedoch mit allen anderen E-Mail-Programmen auch verwendbar. Große Unterschiede gibt es jedoch im Bedienkomfort: Je besser GnuPG in ein E-Mail-Programm integriert ist, desto einfacher die Verwendung.

Die einfachste Methode, z.B. wenn ein E-Mail-Programm überhaupt nichts über GnuPG weiss, ist die Verschlüsselung via Zwischenablage mit Hilfe von Kleopatra. Dies funktioniert nur für OpenPGP, für S/MIME und kompliziertere PGP/MIME E-Mails werden Sie über eine Zwischenspeicherung als Datei gehen müssen. Beide Methoden werden im ersten Teil dieses Kompendiums beschrieben.

Ein Integration in GnuPG wird derzeit für folgende E-Mail-Programme unter Windows angeboten:

Thunderbird mit **Enigmail**[1].

Outlook ab Version 2003 mit GpgOL. GpgOL ist Bestandteil des Gpg4win-Pakets.

Claws Mail: Dieses E-Mail-Programm wird im Gpg4win-Paket mitgeliefert und kann optional installiert werden. Eine solche Installation konfiguriert bereits die Programmerweiterung für die Verwendung von PGP/MIME und S/MIME. Diese Erweiterung verwendet jedoch nicht Kleopatra und bieten daher derzeit nicht denselben Komfort wie es die Outlook-Erweiterung GpgOL bietet.

Kontact: Eine komfortable und erprobte Integration von GnuPG bieten KMail und Kontact. Sie sind für nahezu jedes GNU/Linux-System und neuerdings auch für Windows und MacOS X verfügbar.

[1] http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP

D Automatische Installation von Gpg4win

In diesem Kapitel wird die automatisierte Installation (ohne Benutzerdialoge) erläutert.

In einigen Fällen, wie z.B. für Software-Verteilungssysteme, ist es notwendig, dass die Installation von Gpg4win ohne die Interaktion über Dialoge funktioniert. Um aber trotzdem vorab alle Installationseinstellungen bestimmen zu können, unterstützt Gpg4win das Setzen des Installationspfads auf der Kommandozeile, wie auch eine Steuerungsdatei.

Der Installationspfad kann mit der Option `/D=<PFAD>` angegeben werden, welche als letzte Option auf der Kommandozeile übergeben werden muss. Der Dateiname (hier: `gpg4win.exe` kann je nach Version variieren. Die Groß- / Kleinschreibung bei der Eingabe in der Kommandozeile ist hierbei wichtig. Eventuell sind noch Zugriffsrechte (z.B. lesen und schreiben) auf den Installationsordner zu setzen. Ein Beispiel:

```
gpg4win.exe /D=D:\Programme\Gpg4win
```

Mit der Option `/S` läuft die Installation „still“ (also ohne Dialog) ab. Ohne Angabe von weiteren Parametern, werden alle Voreinstellungen übernommen.

Gpg4win unterstützt auch eine sogenannte Steuerungsdatei. Mit der Option `/C=INIFILE` kann eine Steuerungsdatei (Name endet üblicherweise auf `.ini`) angegeben werden.

Ein weiteres Beispiel:

```
gpg4win.exe /S /C=C:\TEMP\gpg4win.ini
```

Diese `.ini` Datei sollte genau einen Abschnitt `[gpg4win]` enthalten. Dort können diverse Einstellungen vorgenommen werden, darunter absolute Pfadangaben für die zu installierenden Konfigurationsdateien. Relative Pfade, also abhängig vom aktuellem Arbeitsverzeichnis, dürfen hier nicht angegeben werden. Absolute Pfade enthalten den vollständigen Pfad inklusive der Laufwerksangabe. In der Regel sind die Einstellungen dann anzugeben, wenn nicht die Voreinstellung verwendet werden soll. Ausnahmen davon sind im Beispiel auf der nächsten Seite dokumentiert.

Hier ist ein Beispiel für den Inhalt einer Steuerungsdatei, das **alle** erlaubten Schlüsselworte zeigt:

```
[gpg4win]
  ; Installationseinstellungen. Weg- oder leer lassen für
  ; Voreinstellung
  inst_gpgol = true
  inst_gpgex = true
  inst_kleopatra = true
  inst_gpa = true
  inst_claws_mail = false
  inst_compendium_de = true
  inst_man_novice_en = true

  ; Die Stellen, an denen Verknüpfungen erzeugt werden sollen.
  inst_start_menu = true
  inst_desktop = false
  inst_quick_launch_bar = false

  ; Im Gegensatz zu den anderen Optionen überschreibt diese Option
  ; die Einstellung des Benutzers im Installationsassistenten.
  inst_start_menu_folder = Gpg4win

  ; Standard-Konfigurationsdateien.
  gpg.conf = D:\config\gpg-site.conf
  gpg-agent.conf = D:\config\gpg-agent-site.conf
  trustlist.txt = D:\config\trustlist-site.txt
  dirmngr.conf = D:\config\dirmngr-site.conf
  dirmngr_ldapserver.conf = D:\config\dirmngr_ldapserver-site.conf
  scdaemon.conf = D:\config\scdaemon-site.txt
  gpa.conf = D:\config\gpa-site.conf
```

Ein entsprechender Aufruf zur automatischen Installation mit einer Steuerungsdatei `gpg4win.ini` und einem Installationspfad `D:\Programme\Gpg4win` könnte also wie folgt aussehen:

```
  gpg4win.exe /S /C=C:\TEMP\gpg4win.ini /D=D:\Programme\Gpg4win
```

E Umstieg von anderen Programmen

Dieser Abschnitt erläutert Ihnen, wie Sie von anderen GnuPG basierten Programmen auf Gpg4win umsteigen können. Das Installationsprogramm erkennt einige dieser Programme und warnt Sie in diesem Fall.

Generell ist es ratsam, eine vorhandene Installation eines anderen GnuPG basierten Programms zu entfernen, bevor Gpg4win installiert wird. Es ist hier wichtig, die vorhandenen Zertifikate vorher zu sichern.

Der einzige sinnvolle Weg dies zu tun, ist unter Verwendung der im alten Programm vorhandenen Möglichkeiten. Suchen Sie nach einem Menüpunkt um Ihre privaten (geheimen) Zertifikate zu sichern als auch nach einem Menüpunkt um alle vorhandenen öffentlichen Zertifiakte zu sichern. Sichern Sie diese dann in eine oder mehrere Dateien.

Sobald Sie Gpg4win installiert haben, prüfen Sie, ob Ihre alten Zertifikate bereits vorhanden sind. Sie können dies mit Kleopatra oder GPA machen. Sind die Zertifikate schon vorhanden, so entsprach das alte Verschlüsselungssystem bereits den neuen Konventionen zum Speicherort für die Zertifikate und Sie müssen nichts weiter unternehmen.

Wenn die alten Zertifikate nicht erscheinen, so importieren Sie diese einfach aus den erstellten Sicherungsdateien. Lesen Sie hierzu das Kapitel 19.

Falls Ihr altes Kryptographiesystem GPA verwendet, so können Sie die dort vorhandene Backupmöglichkeit benutzen. Diese sollte sehr ähnlich zu der Funktion in der GPA Version aus Gpg4win sein.

Falls Sie keinen anderen Weg finden, Ihre alten Zertifikate wiederzufinden, so suchen Sie bitte mit den Bordmitteln von Windows nach Dateien mit den Namen `secring.gpg` und `pubring.gpg` und importieren diese beiden Dateien mittels Kleopatra[1].

[1] Dies ist nicht der offizielle Weg, funktioniert aber noch mit allen aktuellen GnuPG Versionen.

Migration von Gpg4win-1.1.x nach Gpg4win-2.0.x

Es wird dingend empfohlen zunächst Gpg4win-1.1.x zu deinstallieren bevor anschließend Gpg4win-2.0.x installiert wird.

Technischer Hintergrund

Das Problem bei einer Migration *ohne* Deinstallation von Gpg4win-1.1.x ist folgende Sequenz:

1. Installation von Version X, inkl. Komponente K.
2. Installation von Version X+1, aber Komponente K wird diesmal deselektiert.

Effekt: Die alte Komponente von K bleibt installiert in der Version X.

3. Deinstallation von Version X+1.

Effekt: Die Komponente K in der Version X bleibt auf dem Betriebssystem verwaist zurück.

Mögliche Lösung: Der Installationsassistent ruft die aktuelle Deinstallation für die Komponente K auf, falls diese selektiert war, aber nicht mehr selektiert ist. Diese Lösung wäre allerdings ein erheblicher Programmier- und Testaufwand, da dies nicht von dem Installationsprogramm NSIS einfach unterstützt wird.

Alternative: Installierte Komponenten dürfen bei einem Update nicht deselektiert werden. Der Aufwand wäre vermutlich geringer, dies ist allerdings keine perfekte Lösung.
Technischer Grund: NSIS fehlt ein vollständiges Paketmanagement.

Dies ist eine Beschränkung von Gpg4win seit der ersten Version.

Anmerkung 1: Bei Sprung von 1.1.x auf 2.0.x tritt dieser Fall *immer* ein, da bestimmte Komponenten K nicht mehr existieren, also auf jeden Fall (automatisch) als deselektiert zu betrachten sind.

Anmerkung 2: Im Falle von MSI übernimmt Windows die Aufgabe, nicht mehr verwendete Komponenten zu entfernen. Das bedeutet, dass der MSI-Installationsassistent in dem obigen Szenario korrekt handelt (alte Komponente K in Version X ist nach Schritt 2 nicht mehr auf dem Betriebssystem vorhanden).

F Deinstallation von Gpg4win

Soll Gpg4win deinstalliert werden, dann sollten Sie zunächst alle nicht notwendigen Anwendungen beenden und alle Zertifikate sichern. Falls Sie auf Ihrem Rechner mit eingeschränkten Rechten arbeiten sollten, ist es für die Deinstallation außerdem notwendig mit **Administratorrechten** angemeldet zu sein. Wurde die Installation bereits über Ihr Benutzerkonto durchgeführt, so verfügt es über Administratorrechte.

Wichtig:
Bevor Sie die Deinstallation durchführen, sollten Sie unbedingt Ihre mit GpgOL bearbeiteten E-Mails in Outlook von den GpgOL-Informationen „bereinigen“. Denn: Gpg4win / GpgOL setzt für jede Krypto-E-Mail in Outlook eine bestimmten Markierung. Sie müssen vor der Deinstallation diese Markierung zurücksetzen, damit andere Kryptographiesoftware Ihre E-Mails später korrekt lesen und z.B. entschlüsseln kann.

GpgOL stellt Ihnen für diese **Re-Migration** direkt in Outlook folgende Funktion bereit:
Wählen Sie einen Outlook-E-Mail-Ordner aus, dessen E-Mails Sie zurücksetzen möchten und klicken Sie im Menü von Outlook auf *Extras → GpgOL Eigenschaften aus diesem Ordner entfernen*.

Sie werden darauf hingewiesen, dass GpgOL (für die anschließende Deinstallation) ausgeschaltet wird. Bestätigen Sie die Frage, ob Sie den Ordner von GpgOL befreien wollen, mit *Ja*.
Führen Sie dieses Kommando nun für alle Outlook-Ordner durch.

Nachdem Sie alle Ordner zurückgesetzt haben, beginnen Sie mit der Deinstallation von Gpg4win.

Es gibt zwei Möglichkeiten die Deinstallation auszuführen:

- Einmal mit den Bordmitteln von Microsoft Windows:

  Öffnen Sie *Start→Einstellungen→Systemsteuerung→Software* und wählen Sie dann *GnuPG for Windows* aus.

  Mit dem Knopf [*Entfernen*] deinstallieren Sie alle Gpg4win-Programmkomponenten von Ihrem Betriebssystem.

- Die zweite Möglichkeit zur Deinstallation von Gpg4win bietet Ihnen die ausführbare Datei `gpg4win-uninstall.exe`. Sie wird mit Gpg4win mitgeliefert und liegt im Installationsordner (in der Regel `C:\Programme\GNU\GnuPG\`). Falls Sie bei der Installation einen anderen als den voreingestellten Pfad gewählt hatten, werden Sie das Deinstallationsprogramm an entsprechender Stelle finden.

In beiden Fällen werden alle Dateien von Gpg4win aus dem Installationsordner sowie die Verknüpfungen im Startmenü, Desktop und Schnellstartleiste entfernt.

Nicht gelöscht werden die benutzerspezifischen und systemweiten Anwendungs-Dateiordner mit den Konfigurationseinstellungen:

- Benutzerspezifische GnuPG-Anwendungsdaten
  in `%APPDATA%\gnupg`, das entspricht in der Regel dem Dateiordner:
  `C:\Dokumente und Einstellungen\<Benutzername>\Anwendungsdaten\gnupg\`

  In diesem `gnupg`-Dateiordner befinden sich sämtliche persönlichen GnuPG-Daten, also die persönlichen Zertifikate, Vertrauenseinstellungen und Programmkonfigurationen.
- Systemweite GnuPG-Anwendungsdaten
  in `%COMMON_APPDATA%\GNU`, das entspricht in der Regel dem Dateiordner:
  `C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\`

Deinstallation von Gpg4win-1.1.3

Nach der Deinstallation von Gpg4win-1.1.3 bleiben folgende Dateiordner bzw. Registryschlüssel zurück:

- `%APPDATA%\gnupg`
  (Wichtig: Hier sind Ihre persönlichen privaten und öffentlichen Zertifikat und GnuPG-Einstellungen enthalten.)
- Registryschlüssel:
  `HKLM\Software\GNU\GnuPG` (Wird in der Gpg4win-2.0.0-Installation entfernt.)
  `HKCU\Software\GNU\GPG4Win`
  `HKCU\Software\GNU\GpgOL`
  `HKCU\Software\GPGee`

G Historie

- „GnuPP für Einsteiger“, 1. Auflage März 2002 und
  „GnuPP für Durchblicker“, 1. Auflage März 2002,
  Autoren: Manfred J. Heinze, TextLab text+media
  Beratung: Lutz Zolondz, G-N-U GmbH
  Illustrationen: Karl Bihlmeier, Bihlmeier & Kramer GbR
  Layout: Isabel Kramer, Bihlmeier & Kramer GbR
  Fachtext: Dr. Francis Wray, e-mediate Ltd.
  Redaktion: Ute Bahn, TextLab text+media
  Herausgegeber: Bundesministerium für Wirtschaft und Technologie (BMWi)
  Verfügbar unter `http://www.gnupp.de/pdf/einsteiger.pdf` und
  `http://www.gnupp.de/pdf/durchblicker.pdf`.
- Revidierte nicht-veröffentlichte Version von TextLab text+media.
- „Gpg4win für Einsteiger“ und „Gpg4win für Durchblicker“, Dezember 2005
  Überarbeitung: Werner Koch, g10 Code GmbH
  Herausgegeber: das Gpg4win-Projekt
- Dank der Erlaubnis des BMWi vom 14. November 2007 wurde der unveränderbare Abschnitt „Impressum“ entfernt und an die aktuelle Version angepasst.
- Das „Gpg4win-Kompendium“ fasst „Gpg4win für Einsteiger“ und „Gpg4win für Durchblicker“ zusammen und ist im Jahre 2009 umfassend für Gpg4win2 aktualisiert und ergänzt worden.
  Überarbeitung:
  Emanuel Schütze, Intevation GmbH
  Dr. Jan-Oliver Wagner, Intevation GmbH
  Florian v. Samson, Bundesamt für Sicherheit in der Informationstechnik
  Werner Koch, g10 Code GmbH

H GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's

overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.